

PROJECTS THEORY APPLICATIONS CIRCUITS TECHNOLOGY

NUTS AND VOLTS

www.nutsvolts.com
May 2016

EVERYTHING FOR ELECTRONICS

1959 VACUUM TUBE HEATHKIT *Restored!*
From The Ground Up!
ANALOG COMPUTER



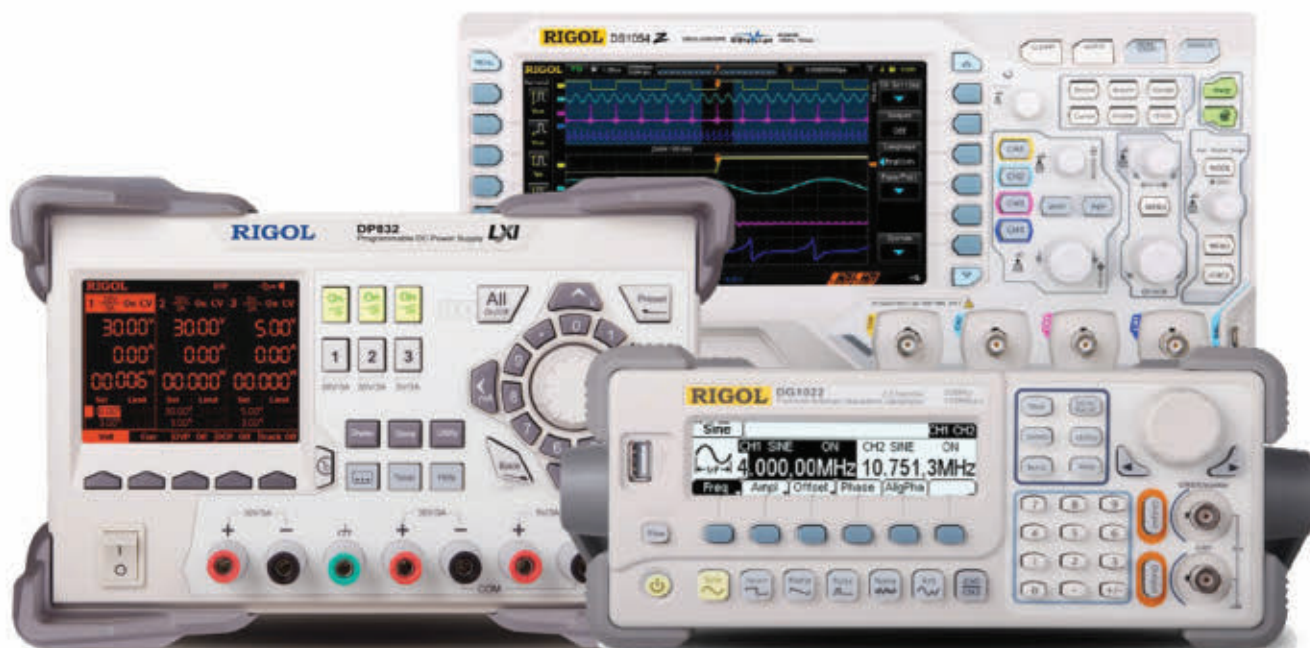
FIVE NEW AMIGO PROJECTS for your mentor's friend
RADIO WAVE propagation explained
RANDOM TONES creating music with math
IR REMOTE SKELETON KEY unlock your volume!

The Future is Here

uncompromised performance, outstanding value

Nuts and Volts Magazine Exclusive Savings!

5% Off Coupon Code: NVMAG



Rigol DS1054Z
50MHz - 4Ch Oscilloscope
\$399

Oscilloscopes



Rigol DP832
Triple Output Power Supply
\$450

Power Supplies



Rigol DG1022
20Mhz Function/Arb Generator
\$379

Generators



"Well designed, easy to use"
Ethan T.



"Tremendous Value"
Nicholas I.



"Hands Down the Best!"
Chris O.



Over 250,000 Satisfied Customers

Call your Sales Engineer for professional solutions and helpful advice.

1-877-571-7901

FREE SHIPPING
on orders over \$99

Over 150 Brands
expert advice

Rigol 30 Day
Money Back Guarantee

RF Specialists

"Making your RF ideas into profitable products."

FCC Part 90 Compliant

USX2 - NBFM Multi-Channel
UHF Transceiver with
Programmable RF Power



MURS (Multi-Use Radio Service)

SHX1- Long Range, High Power
MURS Band Transceiver

VHF NB FM High Power Transmitter

The special HX1 transmitter modules offer a 300mW RF output VHF data link in Radiometrix SIL standard pin-out and footprint.



HF Narrow Band FM Transmitter/Receiver

27MHz High Frequency Narrow Band FM Transmitter
This makes the NTX0/NRX0 pair ideally suited to those low power applications where existing single frequency wideband UHF modules have insufficient range.



Bluetooth / WiFi Combo

Bluetooth Low Energy / WiFi Combination

The LM811 uses Realtek's 8723 BU platform, which enables the 811-0461 HCI Module to be compatible with the latest Android platform, Linux Kernel v4.3.6 and Android 4.4_V2.19.

RF Design Services

Prepared to work with your in-house engineers,
or support your RF project from initial design to implementation.

Industrial • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control



LE MOS

INTERNATIONAL

Tel: 1.866.345.3667
orders@lemosint.com
www.lemosint.com



888-7-SAELIG

www.saelig.com

info@saelig.com



Check out our remarkable, unique products

Test Equipment

Automotive

Cables

Components

Converters

Data Loggers

Displays

Embedded

Industrial PCs

Wireless

Intelligent USB Charge / Sync

cambriox



PowerPad I5



EtherSync

5 Port Smart USB Charger & Intelligent
Multiple USB-Port Charging/Syncing

Best 40 MHz AWG Generator

SIGLENT



SDG2042X

2-Channel 40MHz Function/Arbitrary
Waveform Generator, with 1.2GSa/s
Sampling Rate & 80dB Dynamic Range

Pre-Compliance Test Kit

TEKBOX
DIGITAL SOLUTIONS



TBPS01-TBWA2

High-quality near-field EMC probes
and amplifiers for pre-compliance
testing, LISNs, and TEM cells.

Best 70MHz Oscilloscope

RIGOL
Beyond Measure



DS1074Z Plus

70MHz 4-Channel Oscilloscope, 12 Mpts
Memory Depth, 7" WVGA Waveform
Display, 1GSa/s, MSO Ready!

Best 220 W Triple Power Supply

SIGLENT



SPD3303X-E

DC Power Supply with 5 digits Voltage,
4 digits Current Display and
4.3" True Color TFT- LCD Display

Raspberry Pi - Industrialized!

janzttec



emPC-A/RPI

Fanless Embedded Controller, Based
on Raspberry Pi 2 B Board, Wall or DIN
Rail Mount, CANbus Interface

Free Shipping
Online Orders Over \$99
US Domestic Ground Shipping Only
(HI and AK Excluded)

We also provide the widest selection of PC and RF spectrum analyzers, USB and logic analyzers, waveform generators, digital pattern generators, pure RF sources, DMMs, wireless and wired data-loggers, test equipment, SPI/I2C controllers, touch-sense, Ethernet and USB ICs and modules, CANbus and motor control boards, high-reliability industrial and touch-panel PCs, wireless and I/O control, Ethernet-serial, USB-serial, RS232-485, software, EMI tents and enclosures.

hitec**technology**

hai-tek-nol-uh-jee / noun.

1. Superior servo innovation that is engineered, designed and manufactured exclusively by Hitec RCD. **2.** The highest caliber of servo knowledge and performance for all radio control applications. **3.** Industry-leading research and development, resulting in premium, cutting-edge products. **4.** The combination of all superior Hitec technologies, ensuring the best of the best in servo advancement.



Excellence Defined.

[Manufactured with
hitec*technology*]

Visit us and discover **hitec***technology* for yourself at
booth 1928, AUVSI Xponential 2016, May 2 ~ 5th, New Orleans.

May 2016

24 A Volume Controller Using Any Infrared Remote Control

The sound systems that come standard with today's flat screen TVs leave something to be desired. So, why not fix that and keep your couch potato status at the same time?

■ By Harry Moore

30 1/f Random Tones — Making Fractal Music with an Arduino

Wanna make some music with your Arduino? Then, roll the dice with a unique program that lets you do just that.

■ By Tony Scarpelli

36 The Personal Computer in Amateur Radio

The initial use of computers in ham radio simply involved logging contacts with other ham stations. This discussion will talk about more extensive ham radio contesting programs, with an additional overview of all the myriad uses of computers in ham radio.

■ By John Thompson

10 TechKnowledgey 2016 Events, Advances, and News

Reads this time include nothing to sneeze at, a handy biometric padlock, an opportunity for researchers to get money from DARPA, plus other interesting tidbits.

14 Q&A

Reader Questions Answered Here

Asked and answered is a question on low battery indicators/life, plus a lot of Mail Bag.

20 The Ham's Wireless Workbench

Practical Technology from the Ham World

Waves and Propagation

Radio and wireless are all about waves and how they link distant points together. Learn about RF propagation in the next couple installments of this column.

52 The Design Cycle

Advanced Techniques for Design Engineers

Portable RF Mesh Alarm System Can be Monitored From Anywhere

We will combine an easy to set up and use RF alarm system with a 32-bit IoT node. The result is a portable 40-node RF mesh alarm system that can be monitored from anywhere in the world.

40 Old Heathkits Never Die, They Just Get Restored

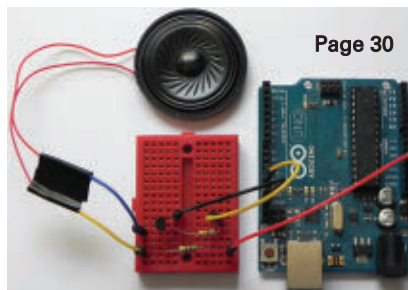
If you enjoy restoring old electronic equipment, then you're gonna love this description of bringing an EC-1 back to life.

■ By David Goodsell

45 Vintage Computing — Five More Projects for the Mentor's Friend

Again utilizing our retro Amigo computer — the home-built mentoring platform discussed originally in our December 2015 issue — here are five additional projects you can do with your young protégé that all help keep the fun and learning going.

■ By Dane Weston



58 The Spin Zone

Adventures in Propeller Programming

Simple. Practical. Propeller.

Whether you create a brand new platform or start with something that already exists, designing your own custom templates for programming projects is really the way to go.

Subscription Information

Nuts & Volts — PO Box 15277

North Hollywood, CA 91615-9218

Call 877-525-2539 or go to www.nutsvolts.com

Subscribe • Gift • Renewal • Change of Info

Departments

08 DEVELOPING PERSPECTIVES

Vinyl Flashback

09 READER FEEDBACK

18 NEW PRODUCTS

18 ELECTRO-NET

19 SHOWCASE

56 NV WEBSTORE

63 CLASSIFIEDS

63 AD INDEX

64 TECH FORUM

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for \$26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615 or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.

NEW

PicoScope[®] 2000 Series PC OSCILLOSCOPES

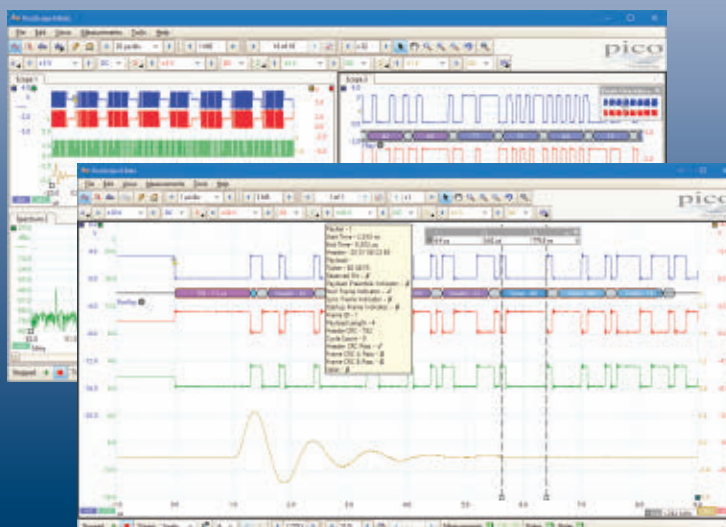
LIKE A BENCHTOP OSCILLOSCOPE,
ONLY SMALLER AND BETTER!

- 2 channel, 4 channel and MSO models
- Up to 100 MHz bandwidth
- Up to 128 MS buffer memory
- Function generator and AWG
- Decode 16 serial protocols
- USB-connected and powered
- Windows, Linux and Mac software
- Prices from \$129 to \$1,235



REDEFINING THE EXPECTATION OF PC SCOPES

All models include full software and 5 year warranty. Software includes measurements, spectrum analyzer, advanced triggers, color persistence, serial decoding (15 protocols including 1-Wire, CAN, Ethernet, I²C, I²S, LIN, RS-232, SENT, SPI, USB 1.1), masks, math channels, all as standard, with FREE updates. Free Software Development Kit also available.



Learn more about this new range at www.picotech.com
or give us a call at 1-800-591-2796 to discuss your application

DEVELOPING PERSPECTIVES

by
Bryan
Bergeron,
Editor

Vinyl Flashback

One of my first jobs in electronics was working in the back room of a RadioShack, repairing whatever customers dropped off — everything from portable transistor radios and car eight-track players, to audio amps and turntables. I especially liked working on turntables — simply slip on a new belt, clean the needle, adjust the speed, and fill in a repair tag. At most, it was a ten minute operation.

Turntables and vinyl records have largely disappeared in the mass market, having first been replaced by CDs and then MP3 files and players. Today, however, there's a resurgence of vintage vinyl, turntables, and old-school tube amps.

I think I've licked the tube amp challenge. As mentioned in a previous editorial, I opted to overhaul a vintage tube amp: a Macintosh 240 dating back to the '60s. A new set of tubes, a dozen capacitors, and a few carbon composition resistors, and the amp is as good as new. Although the specifications are lackluster at best, the sound is perfect for vinyl.

My turntable project presented more of a challenge — mainly because I forgot about all of the peripherals and tools required for basic operation. Operationally, there's an issue of static electricity on vinyl records and associated dust buildup. This calls for at least \$50 of record cleaning fluid, brushes, and related supplies.

Then, there's the issue of mechanical isolation. Despite special shock absorbing feet and foam padding, my turntable skipped whenever anyone walked near it. I relearned that hardwood floors with a bit of spring are horrible for turntables. I tried all sorts of

mechanical isolation products, from spike feet to viscoelastic discs between the player and tabletop. Nothing worked. I finally solved the mechanical problem by mounting the turntable on a wall.

Another challenge was selecting an appropriate preamp. I found a nice tube preamp with great specifications, but only a few gain settings, set via jumper. Because my turntable uses a "hot" pickup, I have to use my amp volume control to vary output power. Next time around, I'm shopping for a preamp with volume control.

Lastly, my task of building a library of vinyl discs resulted in two approaches. If you have the money for new LPs, **Amazon.com** has quite a few vinyl offerings. If nothing else, it's worth checking out the availability of albums on Amazon before you search elsewhere. My go-to "elsewhere" site is **DisCogs.com** — sort of an eBay for music lovers. I've been able to find near mint vinyl LPs for about \$6 each, including shipping.

One of the benefits of vinyl records — also long forgotten on my part — is that the lyrics are printed on either the jacket or record sleeve. Plus, there's the artwork on the record jacket. Contemporary MP3 files with a 16x16 bit icon just can't compete with traditional vinyl record packing.

If you've made the move to vinyl, I'd like to hear about it — especially why you made the move. **NV**

NUTS AND VOLTS

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497

FAX **(951) 371-3052**

Webstore orders only **1-800-783-4624**

www.nutsvolts.com

Subscription Orders

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES

Robin Lemieux

robin@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner

vern@nutsvolts.com

CONTRIBUTING EDITORS

Fred Eady

Jon McPhalen

Ward Silver

John Thompson

Harry Moore

Tim Brown

Jeff Eckert

David Goodsell

Dane Weston

Tony Scarpelli

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEBSTORE MARKETING COVER GRAPHICS

Brian Kirkpatrick

sales@nutsvolts.com

WEBSTORE MANAGER/ PRODUCTION

Sean Lemieux

sean@nutsvolts.com

ADMINISTRATIVE STAFF

Re Gandara

Copyright © 2016 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

READER FEEDBACK

Amped Up Over Tubes

Regarding the April issue's "The Lure of Tube Audio Equipment:" With help from the 1957 ARRL *Radio Amateurs Handbook*, I tackled building a tube audio amplifier. Is it perfectly designed? No. Does it sound good? Yes. Does it make me feel good that I did it myself? Yes. Anyway, I had fun designing and building this 25 watt/channel amplifier (6SN7 preamp tubes and 1625 power amp tubes). The power supply is on a separate chassis. Thought you might like to see a picture. (P.S. I use and program the Texas Instruments' MSP430 microcontroller also.)



James Kretzschmar AE7AX; Cheyenne, WY

James, it's beautiful! Thanks for sharing.
Bryan Bergeron

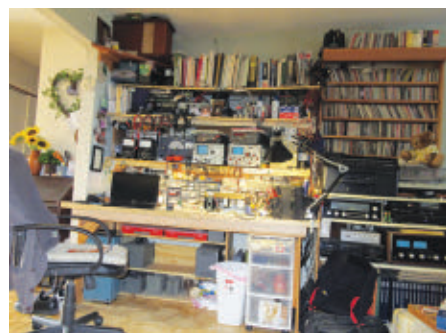
Vintage Work

I've been using vintage repairable electronics for many

years. I bought my McIntosh MC 2105 amp and MX 115 tuner pre-amplifier in 1987. Since then, I have had to replace some output transistors in the power amp twice.

On my workbench now, I have a Zenith Trans-Oceanic L600 which is getting some new caps. My bench scopes are a Tek 485 and a Tek 454 — both of which I can fix. (My first real job was fixing scopes for Philips.) I am thinking of getting another 454. (It's a disease.)

For radio work, I like the Heathkit IG-42 (recapped), though you need to use a frequency counter to make sure these old hollow state generators are at the frequency you think they are. I've included a picture of my bench. My wife lets me keep it in the living room of our condo.



Mike McGinn

Wow! Quite a setup, Mike. I actually know the 454 scope. Used it as a technician way back when. And kudos to your XYL for permitting you to take over some living space with your gear.

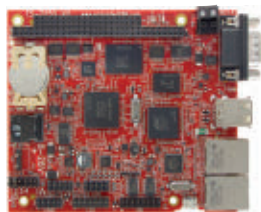
Bryan Bergeron

SUPERIOR EMBEDDED SOLUTIONS



www.embeddedARM.com

Call (480)837-5200



TS-7250-V2 Single Board Computer

Extensible PC/104 Embedded System with Customizable Features and Industrial Temps

- 800 MHz or 1 GHz Marvell PXA166 ARM CPU
- 512 MB DDR2 RAM
- 2 GB SLC eMMC Flash Storage
- PC/104 Connector
- 8k or 17k LUT FPGA
- Dual Ethernet



Available with
TS-ENC720 enclosure

Starting at
\$165
Qty 100
\$199
Qty 1



PC/104 Peripherals Multi-Function Daughter Cards

Our Line of PC/104 Peripheral Boards Offer Maximum Functionality at Minimum Cost.

- Digital I/O Boards
- Ultra Long Range Wireless
- Software Controlled Relays
- Multi-tech GSM, GPRS, HSPA modem boards
- Intelligent battery back-up
- Much more....

Starting at
\$32
Qty 100
\$30
Qty 1



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

ADVANCED TECHNOLOGY

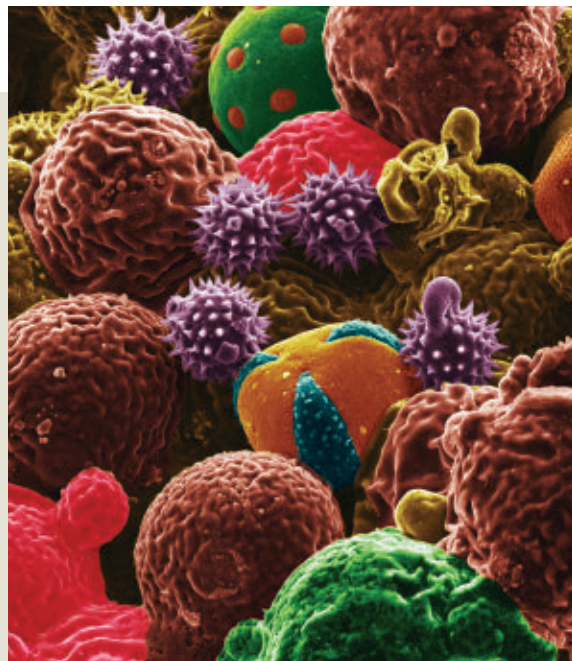
Nothing to Sneeze At

For most of us, pollens are pesky allergens that cause sneezing, itching, runny nose, and other unpleasant reactions. To Prof. Vilas Pol at Purdue University (www.purdue.edu), they are something different. Somehow drawing a connection between his mother's nasal congestion and biomass carbonization processes, Pol recently demonstrated that pollens can be processed into carbon structures that might be useful as anodes in energy storage devices. Pol and doctoral student, Jialiang Tang studied both bee and cattail pollens — both of which are abundant in nature. The main difference is that bee pollen is a mixture of whatever the honeybees happened to collect, whereas the cattail pollens are all the same shape.

Using a process involving pyrolysis in argon gas, they created pure carbon in the shape of the original pollen particles. These were further activated by heating them in the presence of oxygen, thereby forming pores that increased their energy storage capacity. The result was a material that proved useful for anodes in lithium-ion batteries. As documented in a paper appearing in *nature.com's Scientific Reports*, it took 10 hours for the anodes to reach a full charge, but even charging them for only one hour resulted in better than half a full charge.

"The theoretical capacity of graphite is 372 milliamp hours per gram, and we achieved 200 milliamp hours after one hour of charging," according to Pol. "The bottom line here is we wanted to learn something from nature that could be useful in creating better batteries with renewable feedstock."

It turns out that the cattail pollens worked better, but studies continue. Future research will include trying them out in a fuel-cell battery with a commercial cathode. "We are just introducing the fascinating concept here," Pol said. "Further work is needed to determine how practical it might be." ▲



■ SEM image of bee pollen used for study.
Credit: Purdue University/Jialiang Tang, Kay J. Hagen.

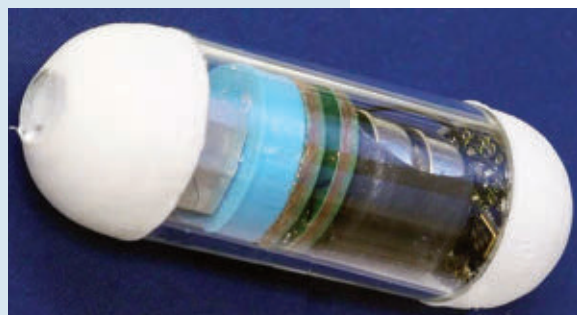
Breakthrough in Pill-Cams

Swallowable pill-cams have been around for several years, providing doctors with close-up diagnostic images of patient's innards. Most rely on the use of small visible-light sources, so clinicians have been restricted to finding only whatever can be detected using the visible spectrum. However, researchers at the University of Glasgow (www.gla.ac.uk) recently came up with a video-pill that uses fluorescence and thereby promises to be more effective at detecting cancers of the throat and gut. Fluorescence has proven to be more effective than visible light in identifying the richer blood supplies that enable cancer cell growth. Existing fluorescence imaging technologies have been too expensive and bulky, and have consumed too much power to be integrated into video-pills. These folks, however, have developed an advanced semiconductor single-pixel imaging technique that is sufficiently compact and energy efficient to image the gastrointestinal tract, and do it for up to 14 hours.

According to Dr. Mohammed Al-Rawhani, "We've confirmed in the lab the ability of the system to image fluorescence 'phantoms' — mixtures of flavins and hemoglobins — which mimic closely how cancers are affected by fluorescence in parts of the body like the intestines, the bowel, and the esophagus. The system could also be used to help track antibodies used to label cancer in the human body, creating a new way to detect cancer."

The device is not quite ready for commercial production, but the university is "in early talks with industry to bring a product to market. We're also interested in expanding the imaging capabilities ... to new areas such as ultrasound in the near future." ▲

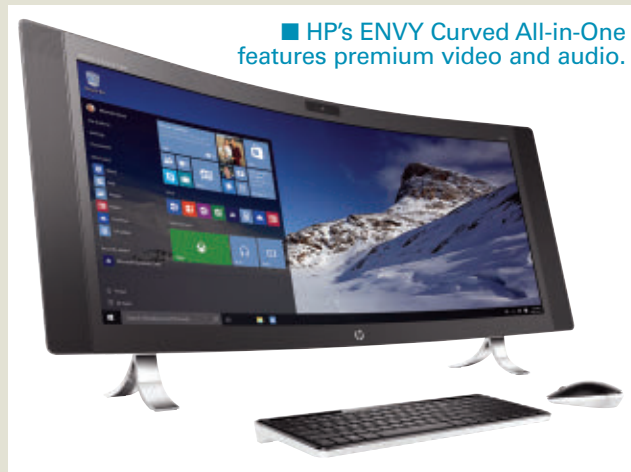
■ Newly developed pill-cam employs fluorescence.



COMPUTERS and NETWORKING

Dual Monitors Not Necessary

Many of us are hooked on having lots of display area and therefore run dual monitors. This works pretty well, but there is that somewhat annoying gap between the two screens, and you have to adjust the viewing angles to maintain a reasonably consistent distance between the images and your eyeballs. Fortunately, both problems are eliminated with the HP ENVY Curved All-in-One: the top dog in a new premium consumer PC lineup from Hewlett Packard (www.hp.com). Billed as the “widest curved all-in-one in the world,” it features a 34 inch diagonal Technicolor Certified display for accurate color reproduction whether you’re watching a movie, processing photos, or just schlepping



around the Internet. It also incorporates Intel's RealSense® technology that employs a 1080p HD camera, an infrared camera, and an infrared laser projector to enable immersive video, 3D capture, and interaction with the PC via gesture control.

On top of that, you get Bang & Olufsen audio with six speakers. In terms of computing guts, there is a choice of Intel's 6th-gen Skylake i5 and i7 processors and an optional NVIDIA GeForce 960A graphics unit for video editing and gaming. Base price is \$1,799.99, with street prices turning up at \$1,649.99. ▲

■ The \$79 Endless Mini comes preloaded with 100+ apps.



PC for the Other 75 Percent

Quite a few computers now populate the under-\$100 world, and one of the latest is the \$79 Endless Mini desktop PC from Endless Mobile (endlessm.com). It differs from the rest of the herd in a couple ways. For one thing, it comes packaged in a fairly attractive case instead of a boring little box or no case at all. The biggest difference, however, is that its intended user base is the 75 percent of the world that still doesn't have (or can't afford) Internet access. Sure, it comes with an RJ-45

Gigabit Ethernet port and a browser for people who do, but each machine is preloaded with an encyclopedia, educational materials, recipes, and other info, plus more than 100 apps, so users can have an Internet-like experience even if their access is slow or nonexistent.

The software may become dated over time, but it can be updated whenever Internet access actually is available. For your \$79, you get an Amlogic S805 quad core ARM Cortex A5 (1.5 GHz), a Mali-450 GPU, 1 GB of RAM, and 24 GB of storage. I/O choices include two USB 2.0 ports on the rear and another one on the front, 3.5 mm stereo output and a mic input, and HDMI video.

If that isn't quite enough, you can opt for a 32 GB version that also includes Wi-Fi and Bluetooth, but that will run you an extra twenty bucks. What you don't get in either case is a monitor, mouse, keyboard, or the ability to run any Windows applications. (They run the proprietary Endless OS.) ▲

INDUSTRY and the PROFESSION

DARPA Challenges Researchers

If you're looking to pick up some extra bucks (up to \$60 million is available), you might check out a recent challenge from the United States Defense Advanced Research Projects Agency (DARPA, www.darpa.mil). The agency has announced a new program that aims to build a connection between the human brain and the digital world. To achieve the goals of its Neural Engineering System Design program, the agency is taking proposals from anyone who can "design, build, demonstrate, and validate a human-computer interface that can record from more than one million neurons and stimulate more than 100 thousand neurons in the brain in real time. The interface must perform continuous, simultaneous full-duplex interaction with at least 1,000 neurons — initially in regions of the human auditory, visual, and somatosensory cortex."

Be forewarned, however, that DARPA wants innovative — not incremental — proposals, and you'll need to incorporate microelectronics, photonics, scalable neural encoding, and processing algorithms. Oh, and the device must be no bigger than two stacked nickels and be secure against spoofing, tampering, and denial-of-service attacks. For details, send your query to DARPA-BAA-16-09@darpa.mil. Good luck! ▲



■ Wanted: human-computer interface.

CIRCUITS and DEVICES

■ The TASCAM DR-22WL portable audio recorder.

Full-Featured Stereo Recorder

For aging audiophiles such as yours truly, the name "TASCAM" conjures up images of 50 lb, eight-channel magnetic tape recorders with 10.5 in reels spinning above a bank of VU meters. "Portable recorder" brings back memories of the little Panasonic cassette machines with a single built-in microphone and truly awful sound quality. However, times have changed, as evidenced by TASCAM's DR-22WL, said to be "the first portable recorder to include Wi-Fi for transport control, file transfer, and audio streaming to your smartphone or PC." It comes fitted with twin condenser microphones arranged in an XY pattern for true stereo imaging, and it includes the company's eight-position Scene Dial which optimizes the recorder for common recording scenarios, offering presets for recording instruments, voiceover, and other subjects.

An EZ mode sets levels automatically, and the manual mode allows you to dial in your own settings. A programmable gain amplifier adjusts audio input levels, making the recorder simple to operate. The unit records in both WAV and MP3 formats, and captures sound pressure levels up to 120 dB. Recording time, naturally, depends on format and sample rate, but it comes with a 4 GB microSD card for storage. Other features are too numerous to mention, but suffice it to say that this is a pretty sophisticated machine for \$150. Details can be found at tascam.com. ▲



CIRCUITS and DEVICES Continued



■ Talon MR60 padlock uses fingerprints instead of keys.

Handy Biometric Padlock

For some time, manufacturers of computers and other high tech equipment have been offering fingerprint recognition as an option, but what about security for file cabinets, tool sheds, and other mundane entities? We've used the trusty old padlocks for years, but it's only a matter of time before we forget the combination or lose the keys. Well, Talon (www.talonbrands.com) has a solution in the form of its MR60-1TB biometric padlock, which can store up to 10 fingerprints.

The weather-resistant lock can be used indoors or out, and it even works in the dark, so you don't need a flashlight. Power is supplied by two AAA cells, which last for up to 4,000 operations, and a low battery warning is provided when enough power remains for only about 100 cycles. A cable version — the MR62-1TB — is also available. The official list price is \$109, but you can pick one up online for about \$89. ▲

Levitating Speaker

■ The ICE Orb Bluetooth speaker floats above its base.

This month's overpriced underperforming gadget that you'll really want is the ICE Orb Floating Speaker from ICE X Electronics (icemobile.in). Based on technology licensed from Levitation Arts, Inc., this Bluetooth 4.1 speaker orb will amaze your friends by floating 10 mm above its base while playing audio from your tablet or smartphone.

Officially, the 5W amplifier covers the frequency range of 40 Hz to 20 kHz, but with a single small speaker, don't expect much on the bass end. You do get up to 10 hr of operation at 60 percent volume, and you can spin the speaker while it plays. Note that it generates a fairly strong magnetic field, so be careful what you put in close proximity. The device gets four out of five stars from reviewers on Amazon, where you can pick one up for \$149.99. **NV**



Your Single Source of Electronic Equipment, Tools, and Components

01DM820B

Digital Multimeter Super Economy

DMM - Volts, current, Resistance, Transistor and Diode Test. Includes Leads

\$7.50



01DMMY64

Digital Multimeter 32 Ranges

Including temperature, capacitance, frequency and diode/transistor testing

\$29.95



01DMM9803R

Bench DMM True RMS

Auto and manual range, RS232C standard interface. Runs on DC or AC

\$139.95



Sweep Function Generator

3MHz, 6 Waveform Functions, Int/Ext Counter, lin/log sweep

Model FG-30 (no display) **\$135**

Model FG-32 (5 digit display) **\$195**

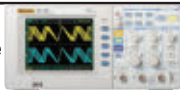


01DS1102E

100MHz Rigol Oscilloscope

Features: 1 million points of deep memory, FFT, record and replay, roll mode, alternate trigger mode, and adjustable trigger sensitivity

\$399



01SDS1052DL

Siglent 50MHz DSO

500MSa/s Sampling 32K Memory 6 Digit Frequency Counter USB Interface 7" Color TFT-LCD Screen

\$279



DC Power Supplies

Model HY3003

Variable Output, 0-30VDC,

0-3Amp.

\$95

Model HY3003-3

Two 0-30VDC, 0-3Amp Variable Outputs plus 5V 3A fixed.

\$195



Tronix Labs

Cool experiments with no soldering.

Includes all Components, Breadboard, Battery, and Manual



Tronix1...Fundamentals

\$29.90

Tronix2.....Digital

\$47.50

Tronix3.....Amps

\$41.95

0603ZD98

48W Soldering Station

Temp. Controlled 310°F-840°F

\$15.25



1 (800) 972-2225

| <http://www.elexp.com>

| contact@elexp.com

In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

● Low Battery Indicator/Battery Life

● Mailbag

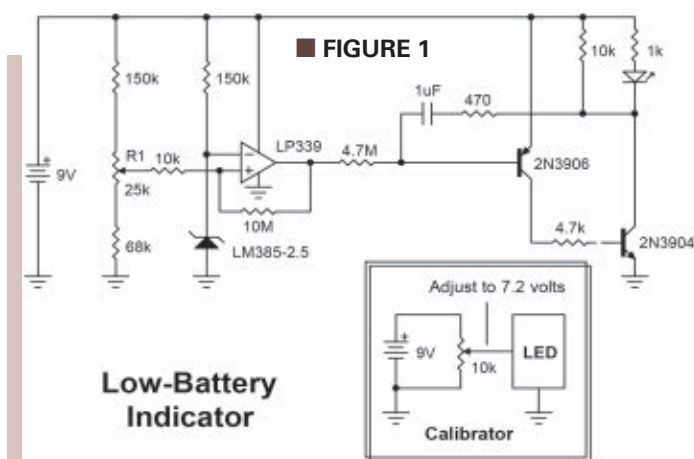
Low Battery Indicator/Battery Life

Q There was a Q&A reprint in a recent *N&V* newsletter titled Low Battery Indicator. Regarding the circuit shown in **figure 1**, it states "The circuit draws a mere 120 microamps, which is just slightly more than the self-discharge current of the same battery sitting on a shelf." I am not sure this is correct.

I calculate that at nine volts the threshold adjustment divider will consume 37 microamps, the reference divider 43 microamps, and from the spec sheet, I_{cc} typical for the comparator is 60 microamps. This all adds up to 140 microamps, which is reasonably close to the 120 microamp figure. According to Duracell, the capacity of an MN1604 battery at low drain rates is about 600 milliamp hours. If you divide 600 mAh by 140 microamps, you come out with a battery lifetime of about 4285.7 hours, which is about a half a year.

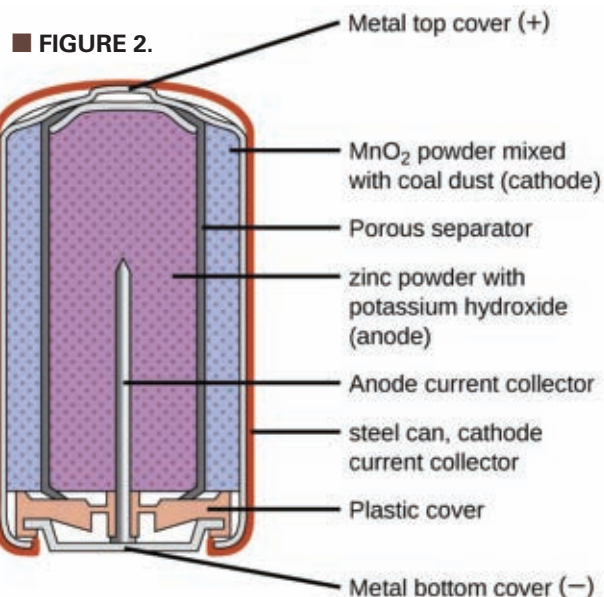
Duracell guarantees a shelf life of five years, after which the battery should still provide acceptable service. So, I think the self-discharge current in the battery must be lower than the current consumed by the circuit by at least a factor of 10 and probably closer to a factor of 50 to 100. Does this all make sense to you or did I screw up my math somewhere?

Vincent Sullivan
North Saanich, BC



A This question and circuit (from the January 16, 2016 *N&V* weekly newsletter) is based on a Q&A column by my predecessor, T.J. Byers. From the Texas Instruments datasheet, I_q (current with IC not sourcing current to load and not in a switched mode) is 25 microamps per channel (we are only using one of the four comparator channels). For the 2N3906 transistor, the collector cutoff current, I_{CCO} , is 50 nanoamps (0.050 microamps) which I will use 0.100 microamps for the two transistors in parallel (the LED will limit the current, but this is a rough estimate).

Adding in the 37 microamps for the threshold adjustment divider and 43 microamps for the reference voltage divider (I calculated the same values you did), the total current when the LED is not illuminated is 105.1 microamps which is lower than Mr. Byers' calculation. However, he may have used a different datasheet or had different operating conditions. Since the original calculation is approximately halfway between your calculation and mine, it is reasonable to use the 120 microamp current draw for the circuit. (See Q&A SIDELINES for links to the datasheets.)

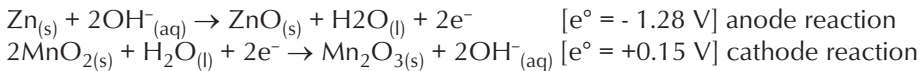


adapted from
http://chemwiki.ucdavis.edu/Textbook_Maps/General_Chemistry_Textbook_Maps/Map%3A_Chemistry_%28OpenSTAX%29/17%3A_Electrochemistry/17.6%3A_Batteries_and_Fuel_Cells

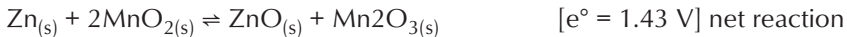
QUESTIONS and ANSWERS

Post comments on this article at
www.nutsvolts.com/magazine/article/May2016-QA

Figure 2 shows a cross section of a single cell of an alkaline battery (the nine volt battery uses six series connected rectangular cells). The chemical reactions that produce energy in the alkaline battery (e° is the electrochemical potential; [s] is a solid material; [aq] is an aqueous solution; [l] is a liquid material) are:



Overall reaction:



When a battery is connected to a load, an electrical current is produced by the voltage potential produced by the chemical reactions in **Figure 2**, as shown in **Figure 3**. When current flows due to the battery's chemical reactions, the original chemicals ($\text{Zn}_{(s)}$ - solid zinc metal and $\text{MnO}_{2(s)}$ solid manganese oxide) are no longer available for further reactions unless the battery is recharged.

Potassium hydroxide (KOH) is the source of the hydroxide ion (OH^-) in the alkaline battery (the potassium is not involved in the reaction; it just brings along the hydroxide ion). Potassium hydroxide is related to sodium hydroxide which we know as lye, which is used in some drain cleaners. Thus, potassium hydroxide is highly corrosive to all of the metals in the battery (zinc anode, steel can, and top and bottom caps). If the KOH corrodes through the steel can, the electrolyte leaks out and ruins the device in which the battery is installed.

In a battery which is sitting on the shelf during storage, the potassium hydroxide electrolyte continues

to react with the zinc anode and manganese oxide cathode, so there are less of them left to produce more energy. As the anode and

cathode material are made inert by these chemical reactions, the voltage of the battery decreases. At some point, the voltage of the battery is too low to power our electronic devices and needs to be disposed of in compliance with local ordinances.

A battery not connected in a circuit does not have a current as we know it (there is an internal flow of electrically charged atoms), so you cannot calculate the current based on the battery's amp-hour rating. Self-discharge is really deterioration of the energy producing internal components of the battery. **Figure 4** shows the service life (time until the terminal voltage is five volts) for the Duracell MN1604 nine volt battery. For the different load currents, I calculate the amp-hour ratings as 250 mA - 2.50, 100 mA - 0.32, and 50 mA - 0.43 which shows that the amp-hour rating is dependent on the load current.

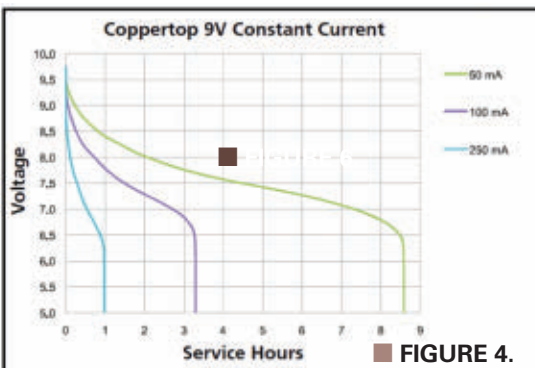
Battery service life is highly dependent on the device it is powering, the load current, ambient temperature, and load cycling.

Tim Brown N&V Q&A

■ **FIGURE 3.**



adapted from <http://www.energizer.com/about-batteries/how-do-batteries-work>



■ **FIGURE 4.**

adapted from http://ww2.duracell.com/media/en-US/pdf/gtcl/Product_Data_Sheet/NA_DATASHEETS/MN1604_6LR61_US_CT.pdf

MAILBAG

Re: Furnace Data Acquisition

#1 Hello, Tim! I enjoyed reading your reply to my comments (Mailbag, February 2016) regarding furnace data acquisition. Thank you for the nice words. I was impressed that you pointed out to your readers that thermostats can differ markedly. In addition to the terminals you mentioned, I have E, L, G1, and G2 terminals, although some are not connected.

With my electric heat pump system, I would definitely want to also monitor contact closure for stage two, which signals the addition of resistance heating coils on extremely cold nights. I would like to contribute two more comments to the discussion regarding your schematic shown in **Figure A**. Following the recommendations given in the datasheet for the 7805 (ST Microelectronics), I would definitely add .1 μF capacitors to ground on the microcontroller side of the regulators. You do not specify the value for C1 and C2, but I would assume you were intending a substantial value electrolytic capacitor. I might consider paralleling those with .33 μF caps based on the datasheet, but perhaps that would be unnecessary and wasteful.

and receivers (sold by Elsema). By FCC regulations, the CB radio frequencies that can be used for other than voice communications are the 27 MHz Industrial Scientific and Medical (ISM) bands (26.957 MHz to 27.283 MHz) which are used by some R/C toys. Maybe readers can help, but I can find no CB radio that allows data inputs and outputs, probably due to the fact that CB is a voice communications device. By obtaining an Amateur (Ham) Radio operator's license, you can use several other bands than the 27 MHz band. (See Q&A SIDELINES for information on Amateur Radio licensing.) In addition to the 27 MHz band FMT-2712 TX and FMT-05D RX devices, Elsema also makes the FMT15101 TX and FMR15101 RX unit for the 151 MHz band (see SIDELINES).

Q&A SIDELINES

Low Battery Indicator/Battery Life

www.fairchildsemi.com/datasheets/2N/2N3906.pdf

www.ti.com/product/LP339

ww2.duracell.com/media/en-US/pdf/gtcl/Product_Data_Sheet/NA_DATASHEETS/MN1604_6LR61_US_CT.pdf

Question about a Previous Q&A Question (R/C Robot)

Amateur Radio Licensing

www.arrl.org/ham-radio-licenses

www.arrl.org/getting-licensed

wireless.fcc.gov/services/index.htm?job=service_home&id=amateur

R/C Transmitter Receiver Modules

www.elsema.com/prod-151.htm

www.elsema.net/pdf/fmr05.pdf

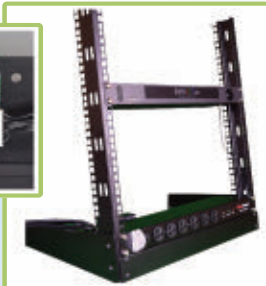
www.elde.cz/datasht/fmt2712.pdf

EARTHLCD.COM
"The Smart LCD Company"

Pi-RAQ

An open source Raspberry Pi Based 1U Rack Mount Internet Appliance With the World's Only 10" x 1"

With the innovation of the Pi-Raq and existing EarthLCD 10x1 display, 1U Rack mount equipment, such as power controllers, audio/video switchers, network hubs and routers, can now have an easily programmable front panel with a color TFT LCD.



Features at a glance:

- 10" x 1" Color TFT
- 1024 x 100 pixels
- 250 Nits
- 7 Function Tact and Scroll Wheel
- 20W Input Power Supply
- 10x1 LCD Adapter Board
- **Raspberry Pi 2**
 - 4 USB Ports
 - 100 MHz Ethernet Port
 - 900 MHz Arm Quad Core CPU
 - 1 GB RAM / 4 GB Micro SD Flash

Not exactly what you need? No problem! EarthLCD can customize a solution just for you!

Contact Us:

For more information or to request a quote
949-248-2333 Direct
sales@earthlcd.com
www.Pi-RAQ.com

All logos and trade marks are respective property of their holder. © Copyright 2015, Earth Computer Technologies, Inc.

The Robot MarketPlace

Supporting Robotic Combat for over 10 years!

- Combat Robot Supplies
- Motors
- Batteries & Chargers
- Vex Robotics
- Carbon Fiber
- Wheels & Tires
- Speed Controllers
- R/C Systems
- Drive Components
- Hobby Supplies & Toys

(877) ROBOT 99
(877-762-6899)
or (941) 749-6030

Your One-Stop Robot Shop!
RobotMarketPlace.com

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

SCOPES DELIVER FUNCTIONS OF SIX TOOLS

Pico Technology's new 2000 Series of oscilloscopes (two-channel, four-channel, and mixed-signal models) have the functionality of an oscilloscope plus a logic analyzer (on MSO models), spectrum analyzer, function generator, arbitrary waveform generator, and serial bus analyzer, with support for 15 protocols included as standard. They

are USB powered and come in an ultra-portable package that can be easily transported in a laptop bag.

PicoScope 2000A two- and four-channel models are ideal for technicians, trainers, students, and hobbyists doing fault finding on signals up to 25 MHz. The 2205A MSO (mixed-signal oscilloscope) has two analog plus 16 digital channels for viewing



ELECTRONET



MaxBotix

High Performance Ultrasonic Rangefinders

www.maxbotix.com
info@maxbotix.com Phone: 218-454-0766

superbrightleds.com

COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



USB

Add USB to your next project--
It's easier than you might think!

USB-FIFO • USB-UART • USB/Microcontroller Boards
 RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com



ALL ELECTRONICS CORPORATION

Electronic Parts and Supplies.
www.allelectronics.com
 Free 96 page catalog 1-800-826-5432



For the ElectroNet
online, go to
www.nutsvolts.com
 click **Electro-Net**

PCB, PCBA and More!

Myro

Low cost High Quality

www.myropcb.com
 1-888-PCB-MYRO





AGENT 390

TRACKED ROBOT KIT
 \$329.99 @ ServoCity.com



Industry's Smallest Footprint

1-800-404-0204
www.ironwoodelectronics.com



Wanna read even more about electronics?
 Check out our book selection at
store.nutsvolts.com

and making measurements on digital or mixed technology designs.

PicoScope 2000B Series models are equipped with deep buffer memory from 32 to 128 M samples, and bandwidths of 50, 70, or 100 MHz supported with sampling speeds to 1 GS/s, and hardware acceleration to deliver over 80,000 waveforms per second update rates. Like the “A” models, they are available with two or four channels, plus 2+16 channel MSO versions. The “B” series models are packaged in the same size ultra-portable enclosure as the A Series, making them ideal for use in the lab or on the move, or for fast low cost shipment to wherever they are needed.

The 2000 Series models connect to a host PC or tablet, and are powered via the USB interface. They are controlled and operated with PicoScope 6 software that takes advantage of available PC processing power, display, touchscreen, and traditional control capabilities. Advanced features such as segmented memory, mask limit testing, advanced waveform math, and decoders for popular serial buses are included as standard.

The new PicoScope 2000A and 2000B Series models range in price from \$129 to \$1,235.

For more information, contact:
Pico Technology
www.picotech.com

HIGH PERFORMANCE SOCKET

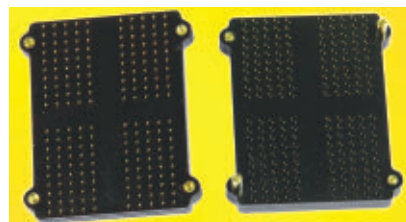
Ironwood Electronics' new high performance socket — the SFS-BGA200B-52 — allows a 0.65 mm pitch, 11x14.5 mm body, 12x22 array 200 ball BGA package to be placed in socket and operated without compromising performance in high speed applications.

The Giga-snaP™ BGA socket adapter pair consists of SFS-BGA200B-52 patented female BGA sockets with BeCu pins assembled into a substrate that matches the male pin adapter.

The RoHS compliant SFS-BGA200B-52 BGA socket is soldered to a printed circuit board (PCB) using standard soldering methods without warping, which results in reliable connection to the PCB. The BGA adapter (to which the user attaches a target 200-pin BGA chip or other compatible chip) is plugged into the female BGA socket on the board; thereby, the chip is interconnected and the system is ready to go.

The Giga-snaP BGA socket adapters require very low insertion and extraction force for ease of operation. The electrical path of the Giga-snaP BGA socket adapters is a high priority performance issue with the physical length from the top connection point on the male adapter to the solder ball on the female socket being 3 mm.

This is a short connection length



for interconnect pin sockets, therefore providing better transmission of high frequency signals up to 20 GHz with -1 dB insertion loss. Operating temperature range is -55°C to +160°C, and the current rating is 3A per pin.

The Giga-snaP BGA socket adapter line is available in many different pin counts/pitches and custom versions can be delivered in days.

Pricing for the SFS-BGA200B-52 is \$276 each, with reduced pricing available depending on quantity required.

For more information, contact:
Ironwood Electronics
www.ironwoodelectronics.com

NATIONAL RF, INC.

TYPE  HFDF
NOISE LOCATION SYSTEM



LOCATE NOISE IN THE HF SPECTRUM
Visit www.NationalRF.com for this
and other Radio Products!
Office: 858-565-1319

We have 33 amazing titles of PROJECTS

FOR THE EVIL GENIUS

Need a project to help you spring into summer?

Check out all the titles on page 57, or visit: store.nutsvolts.com

Only \$19.95 ea

Most popular book!

MORE ELECTRONIC GADGETS FOR THE EVIL GENIUS

40 NEW BUILD-IT-YOURSELF PROJECTS

ROBOT POWER *Extreme Motor Speed Control!*

MegaMoto GT- Mega_Motor Control for Arduino™

- ◆ 6V-40V - 35A / 50A+ peak
- ◆ Single H-bridge or dual half
- ◆ Current and Temp protected
- ◆ Jumper select Enable & PWM
- ◆ Robust power terminals

The VYPER™

- ◆ 8V-42V - 125A / 200A peak!!
- ◆ Single H-bridge
- ◆ RC and Pot control
- ◆ Overload protected
- ◆ For your BIG bots

www.robotpower.com

Phone: 360-515-0691 • sales@robotpower.com

Waves and Propagation: Part 1

Radio and wireless are all about waves and how they link distant points together. Learn about RF propagation in the next couple installments of this column.

One key element of ham radio is *propagation* — the way radio signals get from Point A to Point B. Lots of projects in *Nuts & Volts* involve wireless links, so it behooves the builder to have a bit of an idea about how propagation works. After all, with so many choices available, you need information to understand your requirements for choosing the right technology and installing it properly. We'll start with some basics of radio waves, then get them bouncing around.

Electromagnetic Waves

It would be easy to write a whole magazine's worth of material about electromagnetic (EM) waves and leave the reader more uncertain on the subject than before starting. Let's finesse things a bit by saying that EM energy moves through space as a wave composed of an electric (E) and a magnetic (H) field that oscillate like sine waves at right angles to each other. The energy in the wave is carried by both fields. **Figure 1** shows the E and H fields of an EM wave moving together through space. Note that the fields are not only at right angles to each other, they are at right angles to the wave's motion through space.

The E and H fields don't just happen to be in the same place at the same time. They are *coupled* because they are created by the same thing: an electric charge (usually an electron) that has been terribly inconvenienced from being accelerated by a voltage at the frequency of the wave. Without this acceleration of charge — current in

an antenna, for example — no EM wave is produced. (An animation of the field produced by a moving charge is available at en.wikipedia.org/wiki/Radio_wave.)

The EM wave would be useless for signaling unless we could reverse the process and have the wave cause an electron to accelerate, thus creating a current. That is just what happens when a radio wave is received by an antenna. The E field of the wave creates a voltage in the antenna and that causes the electron to move. If the E field is oscillating back and forth in a sinusoidal pattern, then the electron will do the same. Antennas are bilateral, meaning that they work the same in both "directions." In fact, the whole process is so reversible that according to the math, it can even run backwards in time!

At the transmitting antenna — the one in which voltage accelerates electrons and causes them to radiate EM energy — the E field of the wave radiated by the antenna is aligned with the direction in which the electron is accelerated. Similarly on the receiving end, the antenna needs to be aligned with the E field of the EM wave for the E field to cause the electrons to move in the antenna and produce a detectable current. The alignment of the wave's E field is thus very important, and so its direction is defined as the wave's *polarization*.

If the wave is vertically polarized, then the E field is aligned perpendicularly to the Earth's surface. The E field of a horizontally polarized wave is parallel to the Earth's surface. The polarization of an antenna is determined by the polarization of the EM waves it radiates, so there are horizontally polarized and vertically polarized antennas, too. There are even circularly polarized waves (and antennas) in which the direction of the E and H fields gradually rotates as the wave moves along. Circular polarization has a right-handed and left-handed sense as well, depending on which way the fields rotate.

What's important to remember about polarization is that the receiving antenna and the EM wave must have the same polarization for the wave to create the maximum amount of current in the antenna. If the antenna and wave's E field are at right angles, then instead of the E field causing the electrons to accelerate back and forth along the antenna, the electrons move back and forth across the conductor of the antenna instead of along the antenna. The difference in received signal strength when an

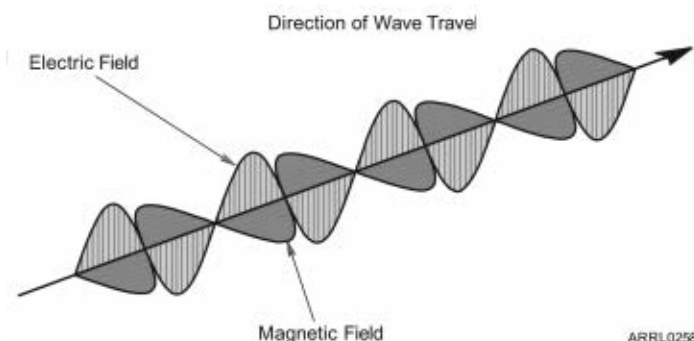


FIGURE 1. The E (electric) and H (magnetic) fields of a vertically polarized electromagnetic (EM) wave moving to the right. The fields vary in the pattern of a sine wave and are oriented at 90 degrees to each other. (Graphic courtesy of the ARRL.)

antenna is *cross-polarized* to the EM wave can be a factor of 100 (20 dB).

Frequency and Wavelength

There is another important part of understanding waves in general and that is the three-way relationship between frequency, velocity, and wavelength. One of those three — frequency, f — stays the same as the wave travels along. Velocity also stays the same as long as the medium in which the wave travels stays the same. That velocity, c , is the speed of light (also an EM wave) in whatever medium the wave finds itself moving through — from the plastic of a coaxial cable to the vacuum of free space. The properties of the medium (primarily its *permittivity*, ϵ , and *permeability*, μ) determine the velocity, with free space having the highest value for $c = 3 \times 10^8$ m/sec. (The velocity in air is only slightly less.)

So, what about wavelength? One wavelength, λ , is the distance traveled by the wave during which the E and H fields undergo one complete cycle of oscillation. Assuming velocity is constant, the faster the wave's fields oscillate, the shorter the wavelength becomes because it takes less time to complete the cycle and vice versa. There is a great animation of acoustic waves that demonstrates this relationship at resource.isvr.soton.ac.uk/spcg/tutorial/tutorial/Tutorial_files/Web-basics-frequency.htm.

The relationship of f , c , and λ is captured in this simple but powerful set of equations:

$$\lambda = \frac{c}{f} \text{ or } f = \frac{c}{\lambda} \text{ or } c = f\lambda$$

For λ to have units of meters, f must be in Hertz (Hz), and c in meters/second.

The relationship can be greatly simplified for calculating the free-space wavelength of an EM wave as:

$$\lambda \text{ (in meters)} = \frac{300}{f \text{ (in MHz)}}$$

If c is lower because of what the wave is moving through, then the wavelength gets shorter, too. When we are talking about the wave moving along a printed circuit board (PCB) trace or a transmission line, this shorter wavelength is called the electrical wavelength. To give you an idea of how much shorter, the velocity of EM waves in coaxial cable with a solid polyethylene dielectric is about

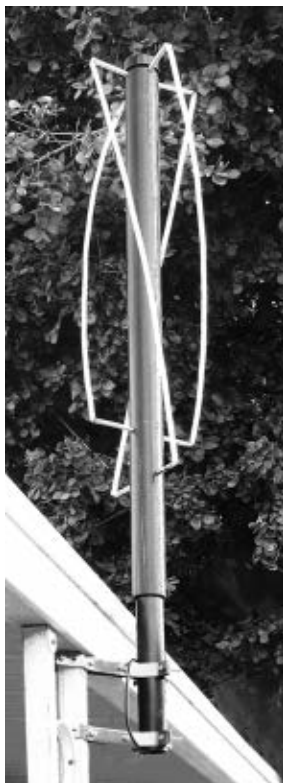


FIGURE 2. This quadrifilar helix built by W6NBC is used for receiving image signals from the NOAA weather satellites at 137 MHz. As an omni-directional antenna, it receives equally well over most the sky. (Photo courtesy of the ARRL.)

two-thirds of the free-space velocity, and the wavelength in coaxial cable is only two-thirds of the free-space wavelength. You have to take this shorter wavelength into account when designing antenna systems, especially.

Omnidirectional and Directional Antennas

Speaking of antennas, we need to launch our waves from an antenna and receive them with an antenna. There are several common types of antennas our wireless systems commonly use, but they are of two basic classes: *omnidirectional* and *directional*. The first type's job is to radiate and receive signals (modulated waves carrying information) in all directions as equally as possible. You met one type of "omni" in this column's first installation which showed how to build a simple quarter-wavelength long whip. That antenna radiated (and received) best in all horizontal directions when held vertically, and so is used on handheld radios, wireless routers, vehicle radios, and so forth. Other types of omnis — such as Lindenblads, turnstiles, and the quadrifilar helix in **Figure 2** — cover a whole hemisphere of the sky and are used for satellite communications.

While an omni antenna is convenient to use because it is relatively unaffected by its orientation, it spreads its signal all over which means less signal at the one point it is really needed: the receive antenna. When a more reliable link with stronger signals are needed

and more transmitter power is not available (or allowed), directional antennas are used. Directional antennas work by focusing the available EM energy in the desired direction. They don't create energy, just focus it; this is a property called *gain*.

Inside our wireless adapter cards and data link modules you might find a *patch* antenna — a type of directional antenna. The patch looks just like its name (see **Figure 3**), and is shaped so that when it is excited by a signal, the electrons flow along specific paths just as if the

Electromagnetic Waves — Going Deeper

This article barely touches the fundamentals of what an EM wave really is: a form of energy traveling through space as time-varying electric and magnetic fields. If you'd like to dig deeper into the fundamentals of EM waves and can handle a little math, the Wikipedia entry on electromagnetic radiation (en.wikipedia.org/wiki/Electromagnetic_radiation) covers the topic well and provides numerous links to supporting pages and resources. EM waves are a fascinating phenomenon, leading to deep questions and answers about the nature of the universe in which we live.

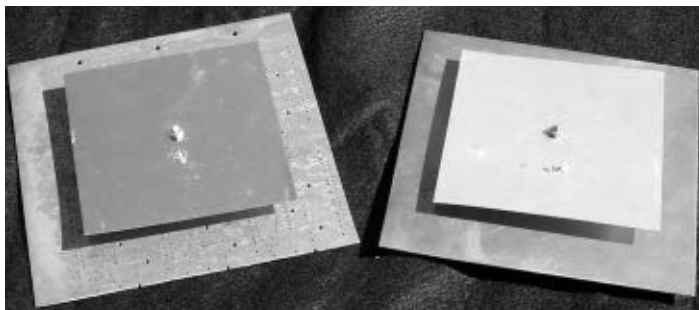


FIGURE 3. These two homemade patch antennas are designed for use at a wavelength of 23 cm (approx. 1.3 GHz). The antennas are one-half wavelength on a side and are excited at their centers. The patch at left is made from sheet metal and on the right from PCB material. (Photo courtesy of the ARRL.)

antenna was made of wires. The net effect is to create a directional antenna that is simple and inexpensive to fabricate for consumer and low cost electronics.

You've probably seen a beam antenna with lots of parallel rods or wires (called elements) along a central support (called a boom). The transmitter feed line is connected to the driven element near one end, and the other elements (reflectors and directors) guide the waves in the preferred direction. The focusing effect can easily increase signal strength at a receiver by 10 times (10 dB) or more. Arrays of beams can be used for even more gain.

The king of the directional antennas, though, is a dish several wavelengths across as shown in **Figure 4**. It uses a curved reflector (the dish) to focus the radio waves from a feed antenna into a tight beam. Dishes can have gains of 30 dB or more depending on how many wavelengths across they are. That is a gain of more than 1,000!

Multipath

If all there was to propagation was picking the right antennas and pointing them at each other, I would simply suggest you buy a copy of the *ARRL Antenna Book* and this column would be done! Obviously, the job can get pretty complicated. At the frequencies we're talking about for unlicensed wireless data links (902 MHz, 2.4 GHz, and 5.6 GHz being the most common), the wavelengths of 33 cm, 13 cm, and 5.4 cm are short enough to be reflected or refracted (bent) by everything from the ground and buildings to desks, windows, and furniture. When a sharp corner or edge is encountered, the signal can be diffracted into a pattern of strong and weak regions. This creates a situation in which the different waves are constructively and destructively interfering with each other.

As an example, when receiving a broadcast FM signal in your car, you've no doubt experienced reception that varies as your vehicle moves along. This occurs because signals from the transmitter are taking different paths to you, arriving in and out of phase with each other. This creates a pattern with alternating areas where the waves



FIGURE 4. This four foot diameter dish antenna is designed for microwave communication in the amateur 2.3 GHz band, but would also work at Wi-Fi's 2.4 GHz. The antenna illuminating the dish's surface is made out of a coffee can! (Photo courtesy of the ARRL.)

add together creating a strong signal, and where they cancel each other creating a "dead spot" or null. As you drive through this pattern, you hear the alternating strong and weak spots as "mobile flutter" or "picket fencing." The peaks and nulls in the signal are about one-half wavelength apart, which for the commercial FM band is about 1-1/2 meters. The next time you pull up at a stop light and your FM station takes a dive into the noise, change the car's position by about four or five feet and I'll bet the signal comes right back!

The same thing happens to your data signal when it encounters a field of obstacles and reflecting surfaces. At the wavelengths for unlicensed data signals, the hot spots or dead spots can be just inches apart. This is called *multipath* propagation and it can cause a data link to be quite unreliable. Wi-Fi routers often try to defeat multipath by using two or more antennas, one-half wavelength apart. This creates two patterns of strong and weak signals that are offset by the half wavelength. This tends to place the peaks of one pattern in the nulls of the other pattern, evening out the signal strength and reducing the number of dead spots.

Scatter

Reflections can also be used to your advantage if they

An Array of Antenna Know-How

The graphics and photos in this article were all taken from the *ARRL's Antenna Book*, now in its 23rd edition (it was first printed in 1939; available at arrrl.org/shop/ARRL-Antenna-Book-23rd-Softcover-Edition). If you're interested in radio antennas and propagation, this is a great book to have on your shelf, with material spanning from 1.8 MHz to 10 GHz. In addition to the *Antenna Book*, there are links to numerous ARRL articles and web pages on antennas and transmission lines and propagation at the ARRL's Tech Portal (arrrl.org/tech-portal). Be advised — set aside plenty of reading and browsing time because antennas and propagation are one of the deepest and most interesting wireless topics.

create a signal path more reliable than the direct path. Or, perhaps the direct path is blocked. In cases like this, it's common to use a directional antenna to find a reflected path around the obstacle or in a direction less affected by multipath. The reflecting surface can be a wall, a building, or even a natural formation. For example, when I lived in Seattle, it was not uncommon to contact someone on a VHF or UHF band by bouncing a signal off Mount Rainier tens of miles away! Not unsurprisingly, this is called *scatter* propagation.

As an interesting experiment, if you have a Wi-Fi beam antenna you can aim in different directions, plug it into a laptop or tablet's antenna port. Open the wireless adapter's configuration software and look for the RSS (received signal strength) indicator. This gives you an indication of how strong the signal is that the adapter is receiving. (Some mobile phones allow you to do this with an external antenna.) Point the antenna in different directions while watching the RSS indicator and note the big variations in signal strength. You will probably find the signal is quite strong in other directions than straight toward your Wi-Fi router or local cell tower.

Scatter propagation does not have to be via a hard surface, either. The scattering surface or structure just needs to be bigger than about one-tenth of a wavelength to reflect a significant fraction of the signal. Hams use all kinds of things to reflect radio signals and increase their range: rain or snow or hail, airplanes flying by, meteor trails that last just a few seconds, and even the moon. If you have ever lost satellite TV service due to heavy rain or snow, the signal from the satellite is being scattered so widely that the amount making it to your antenna is too low to be decoded.

On the Next Transmission

Take a new look at your wireless router. Can you sense the waves boiling off those little antennas? Now you know why they are "that far" apart and also why they are tiltable: so you can change the polarization. The result is to be able to change that pattern of peaks and nulls so you can browse the 'net from that comfy chair in the living room.

Next time, we'll cover things like the radio horizon, effects of the troposphere (that means "weather"), the shortwave bands, and the

Scattering from the Moon and Meteors

You may be quite surprised to learn that actual communication can take place from bouncing radio signals off these celestial reflectors. Such activity is more common today than at any time in amateur radio's history, thanks to software developed by ham radio Nobel Prize winner, Joe Taylor K1JT.

Joe started out in ham radio as a teenager with his brother, experimenting with scatter propagation on the VHF bands. At the Arecibo telescope, he and Russell Hulse developed the necessary expertise to indirectly make the first observation of gravitational waves generated by a binary pulsar. After he returned to amateur radio a decade ago, Joe decided to use that expertise to create software which enabled communication at the extremely low signal levels encountered in "moonbounce" operation and during the short periods of reflection by the ionized trails of meteors.

Joe's software is available as a suite of WSJT protocols (Weak-Signal Joe Taylor), downloadable for free from his Princeton University website at physics.princeton.edu/pulsar/k1jt. All that's needed to get started is an entry-level ham license, a VHF/UHF SSB transceiver, and some surprisingly modest antennas. There are user's groups online to help get started, and you can use the same equipment to operate through ham satellites as well. These are just a few of the fascinating ways for amateurs to interact with the natural world in unexpected ways.

ionosphere, plus some really neat tools for assessing and studying propagation. See you then! **NV**



EARN MORE MONEY Get your dream job!

Be an FCC Licensed Wireless Technician!

Make up to \$100,000 a year and more with NO college degree

Get your "FCC Commercial License" with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- **MONEY BACK GUARANTEE:** You get your FCC License or money refunded!



Your "ticket" to 1,000s of high paying jobs in Radio-TV, Communications, Avionics, Radar, Maritime & more.

Call for **FREE** info: **800-877-8433** ext. 109

or, visit: **www.LicenseTraining.com**

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



A Volume Controller Using Any Infrared

REMOTE CONTROL

By Harry Moore

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/May2016_IR_Volume_Remote_Control.



Here was my dilemma. Out of the box, TV sound is lousy. I suppose it has to be that way, since there is little room for a decent set of speakers in a modern flat screen TV.

When my son (who was working in a big box electronics store) bought me one of these new digital TVs for my birthday a few years back, I was overwhelmed with the picture. With just a little outside UHF antenna, I was picking up snow free/ghost free high definition pictures. I thought it was better than the jittery 35 mm film movies at theaters in those days (though now they are digital, too.)

So, what could I do about the sound? I had several older stereo receivers lying around (remember the old Sherwood, Pioneer, and Harman-Kardon boxes) and a spare (and cheap) computer sound system that had a separate bass box that I used with my old TV set. Any and all of these worked great with the new TV. They provided rich "bassy" three-dimensional sound compared to the tinny flat audio directly out of the set.

Only one problem: I don't consider myself much of a couch potato, but modern (if you consider the last 40 years) remote controls have spoiled me. My improved sound system required repeated trips to the set to adjust the volume as the program demanded (dramatic music too loud, voices too soft).

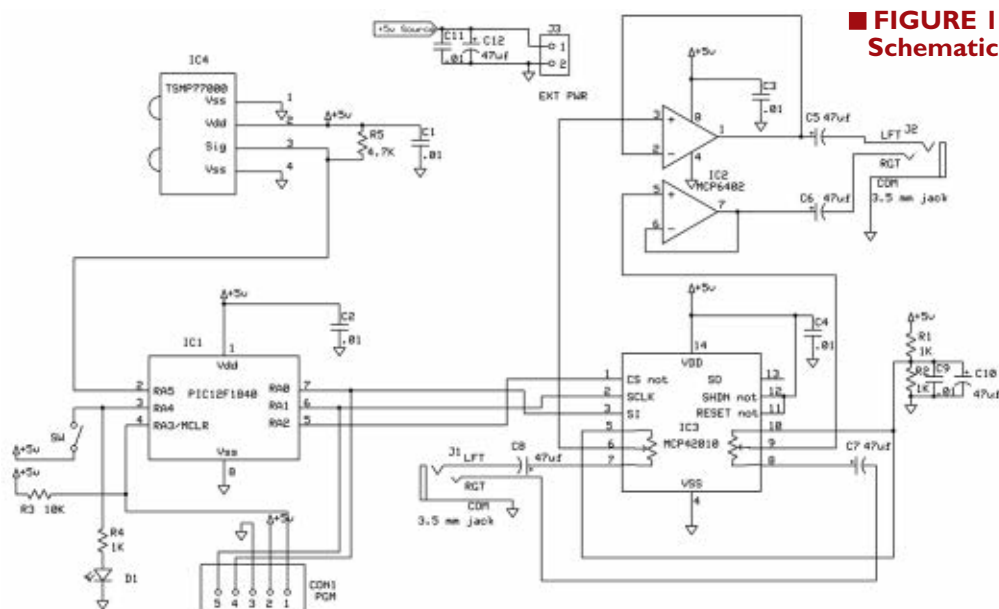
This was because a TV's remote control only adjusts the in-board sound, not that out the audio ports in the back. This makes sense for the TV manufacturer as they correctly assume that you would want the TV speakers to always be turned all the way down when hooked to an exterior sound system.

I also had a 100 disc CD player plugged into one of my 'legacy' receivers: a Yamaha built before remote controls were common. It also gave me my exercise, requiring repeated trips to twiddle the volume control.

What I needed was a simple little box that would work with any hand infrared control (how many of these do you have lying around in your house?) that could be programed to control the audio out of the

TV or CD player (line level). It should even respond to any unused buttons on my TV's existing handheld remote. Plus, there were lots of these mystery buttons that I had no idea as to what they were intended for. This box should plug in between my external sound system and the output jacks on my TV or CD player.

This box should use very little power just sitting there, and be powered through a cell phone size USB jack on the back (any cell phone charger will do). It should have good range (15 feet or so) and be easy to 'train' with infrared commands.



■ **FIGURE 1.**
Schematic.

The Solution

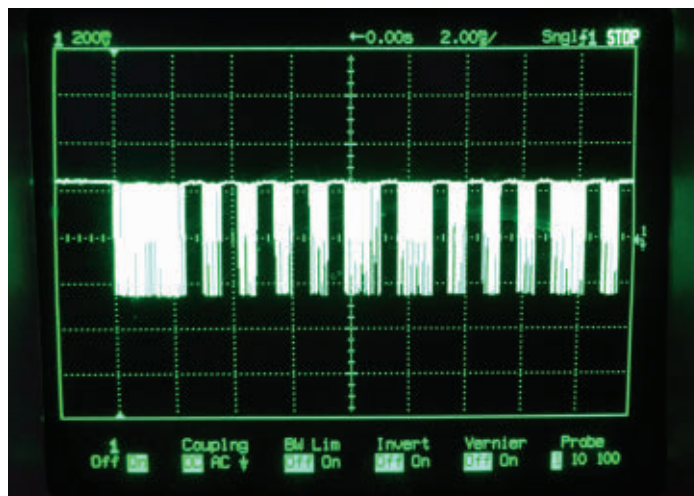
Take a look at **Figure 1** to find what I came up with (I call it the IR1). As you can see, it looks pretty simple: three ICs, an infrared receiver chip, and a handful of passive parts. The 'smarts' are in the PIC12F1840: an eight-pin (cheap) microcontroller from Microchip. **Figure 2** is a view of my first prototype; note the PCB (printed circuit board) from SchmartBoard to aid in attaching the surface-mount ICs.

You may wonder why I prototype with surface-mount parts. Years ago, I admit that I was as resistant as anyone to using surface-mount parts for small run and prototype projects. As time passed, however, it became increasingly obvious that lots of stuff was no longer offered in through-hole versions. It was time to bite the bullet and clean off my glasses.

An additional perk I found was that I could build really small (but not too small) projects and save money on enclosures and PCBs. By sticking to SOIC and TSOP size ICs (50 mil or 1.27 mm between pins) and no smaller than 603 size passive parts (60 mil), I could get away with hand soldering (not with too much coffee, though). Refer to the Surface Mount Aids list for some helpful tools for installing surface-mount parts and a link to a helpful video, as well.

How Infrared (IR) Signaling Works

Your handheld remote control sends out a train of ultrasonic carrier pulses (displayed on my scope in **Figure 2**). This is commonly called pulse code modulation. (See www.vishay.com/docs/80071/dataform.pdf.) The carrier typically runs at 38 kHz to 40 kHz. There are various



■ **FIGURE 2.** Part of an infrared waveform.
Note the start pulse at the beginning.

modulation schemes — some depending on the length, timing, or polarity of the leading edge of each pulse making up each IR command. In our application, we want to work with any IR remote control regardless of the modulation scheme. To accomplish this, we must simply detect the carrier frequency and the timing of the command pulses. Then, we look for a match with a 'trained' signal captured in training mode. We don't care what the underlying format is.

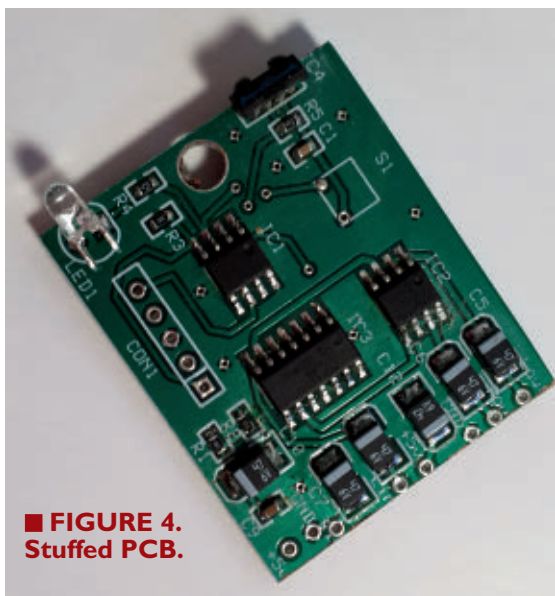
How It Works

The unique hardware that makes this all work includes: IC4, the Vishay infrared receiver; and IC3, the Microchip digital potentiometer.

The infrared receiver does an excellent job of filtering



■ **FIGURE 3. Prototype with cover removed.**



■ **FIGURE 4. Stuffed PCB.**



out noise and responding to low level IR signals. At first, I tried building my own IR receiver with an IR diode detector, but it quickly became too complicated to be practical. The sensitivity and noise rejection requirements are just too steep. The Vishay receiver has a respectable distance signaling spec of 15 feet too.

The Microchip digital potentiometer is the key to controlling the IR1's audio level. It responds to an SPI format digital signal — available from almost any microcontroller — to step the audio level up and down as required. A Microchip op-amp buffers the output of the digital potentiometer before sending it to the outside world.

The Microcontroller

With so many choices between manufacturers, product lines, and features, choosing a microcontroller can be daunting. For this project, I could at least zero in on what series I wanted to use, having worked with the PIC series for years and realizing that their eight-bit versions were easily up to the task at hand.

Processing speed was one consideration; their 32 MHz units — using the internal PLL to bump up the internal 8 MHz clock — also simplified my design (no external crystal/ceramic required). This frequency accuracy was adequate since the IR detection algorithm was self-correcting. If the IR carrier frequency was a bit off, it was off by the same amount for when both programming and detecting an IR signal (as long as there was not too much frequency drift). The timer in these microcontrollers has a max counting frequency of one fourth of the instruction clock or 8 MHz, which is 200X faster than a typical IR carrier of 40 kHz. About one half percent accuracy in measuring the carrier frequency and period should be plenty good.

I settled on the PIC18F1840 with 4K of program memory (plenty) and 256 of EEPROM nonvolatile memory. I would need to store 80 bytes at the most for each of three commands: volume up, down, and toggle mute. Each byte would hold the number of carrier pulses for each of the command's carrier on or carrier off states. Not the most efficient storage method, but so what. I don't need much.

One thing I liked the most was the fact that I only needed to deal with an eight-pin package with this microcontroller, making soldering easier and the PCB size smaller. I did, however, have to make the training select button input do double-duty with the LED on an input pin, which added a little complexity to the code.

The Microcontroller Code

The code for this project is available at the article link. What follows is a brief description of how it works.

Pressing the training button starts the controller sampling the IR code start pulse to determine the carrier frequency. This start pulse is not part of the signaling code and gives us the opportunity to measure the carrier period. Sixteen IR transitions are counted in the start pulse; from this, the carrier period (1/16 the time measured with an onboard timer) is calculated.

With the start of the actual code sequence, the number of carrier on and carrier off pulses in each segment of the IR command are

■ **FIGURE 5. Top view of the enclosure.**

counted and saved in the nonvolatile EEPROM on the microcontroller. Knowing the IR period allows us to accurately count this number of IR carrier pulses. Noise or a different carrier frequency is rejected because the carrier frequency and therefore the period would be incorrect.

Once 'trained' and in operation, the same logic is reapplied with each received IR signal. The incoming carrier period is compared and the incoming pulse sequence is compared to that which is saved in the microcontroller's EEPROM. If there is a match, the microcontroller sends an SPI format signal to the digital potentiometer IC to bump up, down, or turn on or off the audio.

After each volume change or toggling the mute, the current volume setting is saved in EEPROM so that the volume setting will be the same when coming up from a loss of power.

Build It

I've included (at the article link) the files needed to build a very small (I wouldn't go so far as tiny) controller with all surface-mount components. The Parts List provided is for this version. If size is not so important and you find surface-mount work challenging, you can build a version like my prototype which uses many through-hole components; just choose through-hole components for the passive parts and DIP versions of some of the ICs if they are still available (I find this can change quickly).

I've also included a schematic file (IR1.sch),

Working with Surface-Mount Parts on the Cheap

The one indispensable item for surface-mounting parts by hand is a good set of tweezers (see All Electronics website). Regular long nosed pliers are just too big to maneuver the tiny parts to place on your PCB.

Another low cost item is a pair of helping hands w/magnifier (also at All Electronics.) This stand-mounted magnifier can serve as a low cost microscope for placing and soldering ICs and passives to your PCB.

For a little more (less than \$130 from AmScope), a true stereo microscope (with at least a 10X range) is a big asset (also good for checking out bugs).

Some small diameter solder is a must. I use .015" diameter. Solder braid is one of the best ways to remove solder bridges across the pins of ICs.

A 1/32" screwdriver tip on your soldering pencil and some solder flux and paste helps complete the bill to work with surface-mount parts.

Finally, check out PanaVise (<http://panavise.com>), who is a major manufacturer of precision vises, circuit board holders, and work holding tools.

a PCB layout file (IR1.pcb), and two files (IR_rear-panel-holes.pcb and IR_front_panel-holes.pcb) for hole location to use as stencils for drilling the enclosure specified in the **Parts List**.

Note that there is a small daughterboard that holds the micro USB connector for a power input. You can see

ITEM	PART	SOURCE
Stereo Microscope	Sharp SE303-PY	AmScope.com
Solder Paste	SMD291AX	Digi-Key Corp
Solder Flux	SMD291NL	"
.031" Soldering Tip*	ETH	"
.015" Solder	82-105	"
Solder Braid	CW2-5	"
Tweezers		Multiple sources
Hands-free Magnifier		Multiple sources
Soldering Help		http://store.curiousinventor.com/guides/Surface_Mount_Soldering/101

**SURFACE
MOUNT
AIDS**

*Note that this tip is for a Weller Model WE551; select a similiar tip for your soldering station.

PART #	DESCRIPTION	DESCRIPTOR	PARTS/BOARD	SOURCE
TSMP77000	IR Detector	IC4	1	Mouser Electronics
PIC12F1840	uProcessor	IC1	1	"
MCP6402	Op-amp	IC2	1	"
MCP42010	Prog Pot	IC3	1	"
STPX-3501-3C	3.5 mm Jack	J1, J2	2	"
10118193-0001LF	uUSB B Jack	J3	1	"
F930J476MBA	47 µf Cap	C5, C6, C7, C8, C10, C12	6	"
WP710A10QBC/D	Blue LED	D1	1	"
CRCW060310KFKEA	10K Resistor	R3	1	"
CRCW0603XXXXFKEA	4.7K Resistor	R5	1	"
CRCW0603XXXXFKEA	1K Resistor	R1, R2, R4	3	"
GRM188R70J103KA01D	.1 µf Cap	C1, C2, C3, C4, C9	5	"
EVQ-PLHA15	Tactile Switch	SW	1	"
609-3272	Pgm Socket	CON1	1	"
IR1 PCBs	Main and Pwr PCB		1	View and modify on ExpressPCB site.
075X12/BK	Enclosure		1	Standard Injection Molding Co

**IR1
PARTS**



■ **FIGURE 6. Rear view.**



■ **FIGURE 7. Bottom view. Note the hole through which the training button on the PCB is accessed.**

Training the Unit for Your Remote Control

To get the unit up and running, press the recessed button on IR1 and note that the front LED lights to signify you're entering training mode.

Hold your IR remote in front of IR1 and press the button that you want to use for volume up.

Note that the front LED lights in a fraction of a second to indicate that the command was received.

Press another button on your remote for volume down; note that the front LED lights. Finally, press a last button for toggling the sound mute. Again, the LED will light to signify that the command was received.

You're now in operating mode. Try the three buttons on your remote and notice that each time, the front LED flashes to indicate reception of the command. Many remotes will send out continuous commands (the LED stays on as long as you press the button), while some only send one or two at a time (with only a single flash of the LED).

Why is the Code in Assembly?

You may be wondering why the microcontroller code is in assembly. I know that many consider the C language to be a better choice. C is considered a 'portable' assembly language and a universal language across all microprocessor/microcontroller choices.

Some reasons come to mind for using assembly:

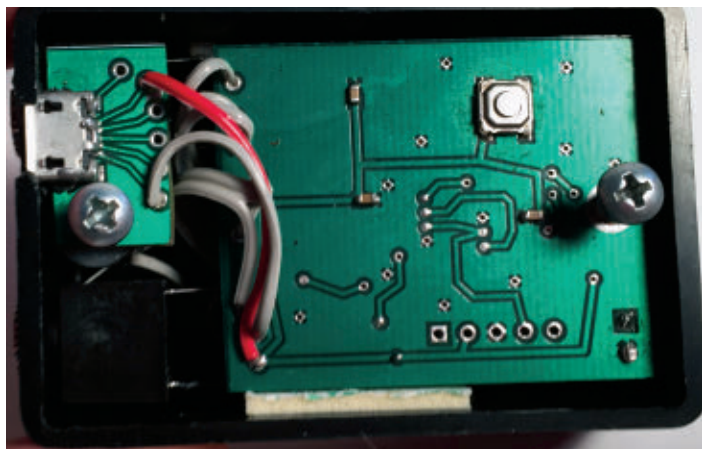
1. In dealing with real time and relatively fast operations, it pays to know exactly the number of instruction clock pulses that occur in any procedure. You can intimately adjust the code to make a procedure as fast as possible, and you know exactly what is going on inside the microcontroller, which helps with troubleshooting and fine-tuning.

2. In addition to its CPU, each microcontroller is made up of a particular kind and number of hardware modules: timers, communication portals, comparators, and

in **Figure 8** how this daughterboard mounts over the 3.5 mm audio jacks. The rear enclosure cover screw clamps this board in place.

All of these files were generated with the free ExpressPCB program (www.expresspcb.com). I've found that their free schematic and layout tools are excellent (and can be easier to learn and use than some high priced commercial offerings). In addition, it is very easy to upload your finished PCB file for manufacturing from the board layout program. I "panelized" (combined several circuits on one board) the layout using their Mini Board Pro option in order to save money. (Four main PCBs and five daughterboards are generated for each of the three panelized boards.)

You can print out the hole location files with the ExpressPCB program and tape the printout over the front and rear of the recommended enclosure. Then, simply center punch or drill through these stencils for the correct hole placements. Because I don't have a custom punch, I used a small circular file to open up the area between the holes for the micro USB power jack.



■ **FIGURE 8. View with the cover removed showing the training button on back of the main PCB.**

the like. The advantage of the C language portability is lost when these different various hardware configurations depend on a unique set of configuration and operating parameters. C just makes this job harder.

3. I just enjoy programming and knowing what is going on in the logic of the microcontroller. There is nothing between me and the hardware. There's something satisfying in knowing exactly what the hardware is up to and manipulating the program at its most 'intimate' level.

Other Uses

In addition to controlling my TV's volume, I control the audio from my older Yamaha receiver that's plugged into a multi disc CD changer. I can now adjust the audio as I like from across the room. I just grabbed an old remote from a defunct TV to use as the IR controller.

Now, if I can just remember where I left that remote.

NV

Resources

For some background on infrared command formats:

www.vishay.com/docs/82478/tsmp77000.pdf

For parts for the IR1 (for the specifications for each part, follow the links on Mouser's website):

www.mouser.com

For the enclosure that's sized for the PCB (part #075X12/BK):

www.simcobox.com

To view, modify, and/or print the PCB, schematic, and hole location files:

www.expresspcb.com

A link to stereo microscopes for help with surface-mount soldering (I use the Model SE303-PY):

www.amscope.com

For miscellaneous surface-mount aids:
allelectronics.com
digikey.com

For easy-to-solder surface-mount prototype printed circuit boards:

www.schmartboard.com

A PCB and programmed chip to go with this article can be purchased online from the **Nuts & Volts** Webstore at store.nutsvolts.com or call our order desk at 800-783-4624.

HF-AUTO High-Power Analog Autotuner



“ My HF-AUTO is a superb piece of engineering magic
W9OWO Dec 2012

“ Beyond awesome I will never sell this fast, accurate, reliable
W2PP Sept 2015

“ Of the 30+ tuners I have owned . . . the HF-AUTO is the best
K1ESE Feb 2012

The HF-AUTO is a microprocessor controlled fully automatic stand-alone tuner with a power rating of 5 Watts to 1800 Watts that will work with any transmitter built from the 1940s to the present. HF Bands: 160m to 6m. Three antenna outputs: SO239 coax. Dimensions: 12.5" W x 6.5" H x 16.5" D. Weight: 25 lbs. (11.4 kg).



HF-AUTO-R Remote unit
Operate from 500' distance

Palstar product reviews from actual Hams as found on eHam.net

PALSTAR

Customer Support: 1-800-773-7931
Fax: 937-773-8003

www.palstar.com

1/f Random Tones — Making Fractal Music WITH AN ARDUINO

By Tony Scarpelli

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/May2016_Fractal_Music_with_Arduino.



Back in 1973, I was looking at a wall. The writing on it said to learn about computers. So, I took a course in computer repair and never looked back.

My wife bought me a KIM-1 single board computer (<https://en.wikipedia.org/wiki/KIM-1>) for my birthday in 1976; it cost \$245. I wrote a KIM-1 program in assembly language that played 1/f random notes. I wrote an article about it, and in July 1976 it was published in *Personal Computing* called, “1/f Random Tones.” They paid me \$400. It paid for the computer for sure.

During the rest of the ‘70s, I wrote a lot of programs for it and began writing articles about the programs. The KIM-1 had 1K of RAM and 2K of ROM. It stored programs on cassette tape, had a six-character display, and you had to program it in assembly language using hex numbers with an ASCII keypad for entry.

Since then, I’ve published many more money-making articles; at that time, the computer magazines paid pretty well. After a couple years, I bought boards to make the KIM-1 a little smarter which allowed me to program in BASIC; that would have been “Microsoft Cassette BASIC.” I was having a problem with it

and called Microsoft. I actually got to talk to Bill Gates on the phone at the time. What did he have to say: “Call your vendor.”

I eventually updated my computer life by getting a Tandy/RadioShack TRS-80 (<https://en.wikipedia.org/wiki/TRS-80>). I built an IBM Selectric (https://en.wikipedia.org/wiki/IBM_Selectric_typewriter) Terminal printer interface and printer driver that was published in a British magazine called *Wireless World*. The Selectric had solenoids in it to drive all the keys. Thus, I was able to print all my code and other writings out in letter quality print. It was slow, but beautiful. There were many, many computer magazines on the store racks at the time and I wrote articles for many of them. Back in 1980, *Nuts & Volts* was printed on news stock and actually looked like a newspaper.

Since we don’t have to use ancient computers any more in our programming, I decided to write the 1/f program using an Arduino. However, let’s first talk a bit about 1/f music.

For a long time, fine arts have been duplicating nature, or at least trying to imitate her as much as possible. Yet, music seems to be the least imitative of those arts. So, how can it be connected to nature's seemingly structured randomness? Well, a certain statistical property of the world appears to be the connection. This property was discovered by Richard F. Voss, an IBM physicist. His discovery concerns the relation or "autocorrelation" between vibrations and their power spectrum.

To understand this concept, we have to consider types of random sounds. For instance, changing the speed at which you play music on your old phonograph naturally decreases or increases the sound's pitch. However, a type of sound called "scaling noise" sounds the same no matter what speed you play it at. An example is white noise like the random noise produced in a resistor, or even plain static. One bit of noise is completely unrelated to the last bit or any future bit. Its autocorrelation factor is zero. You can write a program to generate such random notes, but it soon becomes boring.

A more correlated noise called Brownian noise is also random, but each bit of noise is related to the last bit and to the next bit. To get a picture of this type of noise, imagine a butterfly flying. Its path is apparently random, yet it is connected, though in a wandering flight. Although music made from Brownian noise has a high autocorrelation factor, it still tends to be dull.

Halfway between white and Brownian noise is Voss' discovery, or $1/f$ noise. If white noise is $1/f^0$ and Brownian noise is $1/f^2$, halfway is naturally $1/f$, or pretty close to it. Basing music on this type of noise is a lot more fun and interesting.

Before I get to that, I should explain the term "fractals." Benoit B. Mandelbrot coined the term to cover a class of patterns having the property that no matter how closely you look at them, they always look the same. He discovered that the flooding of the Nile, variations in sunspots, and undersea currents are based on $1/f$ fluctuations. Voss said our total experience is based on $1/f$ noise.

An article by Martin Gardner in *Scientific American* (April 1978) contained an example showing how to produce $1/f$ numbers using dice. I took that example and programmed the KIM-1 to do it repeatedly, and to play notes and tunes based on the $1/f$ numbers. This time, I used an Arduino.

How the Program Works

Say we get three dice, or rather program our

	COLUMN		
ROW	3	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

■ **FIGURE 1.**
Column change
table.

computer to get them. We also make three columns of binary numbers with each column representing one of the dice. Since there are three dice, we need to count in binary up to 2^3 , or eight. Refer to **Figure 1**.

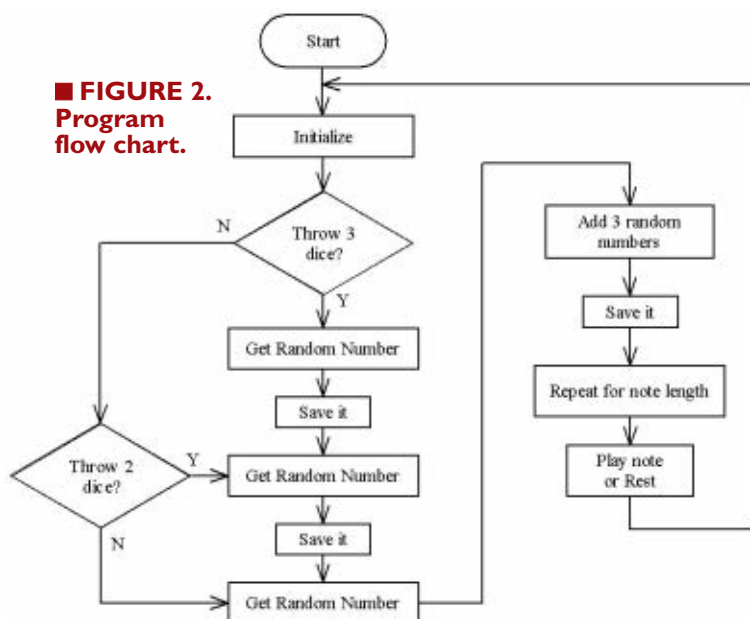
To determine what row is to be thrown, we have a sequence of eight numbers (3, 1, 2, 1, 3, 1, 2, 1) stored in an array called *NumColsToChangeN*. Since the first number is 3, we throw all three dice. The three dice (actually generated random numbers) are added up; this sum points to a note on a piano or, in our case, to a note in memory in another array.

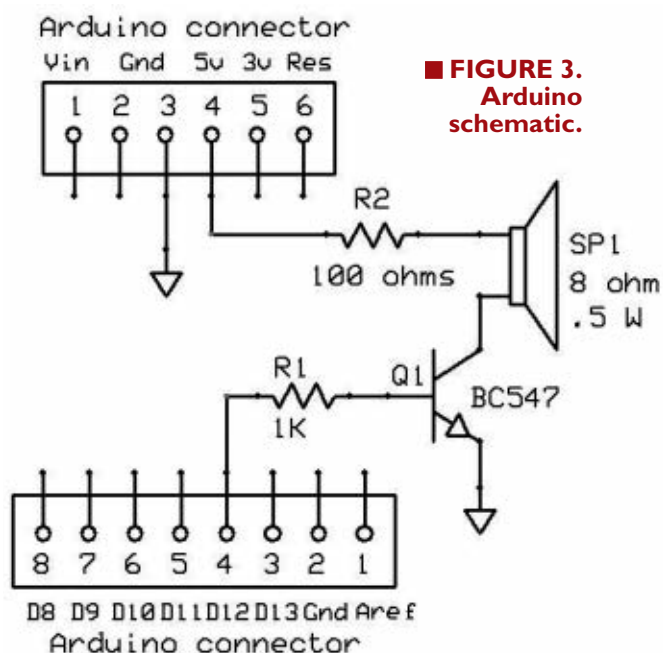
The second number in the *NumColsToChangeN* array is 1, so we throw one dice in the one column. After we throw the one dice, all three dice are added up again, and we point to another note. The next number is 2, so two dice are thrown and again all three are added.

This process continues until we have a pattern for the eight rows. After row seven, the whole thing repeats. The numbers generated are random, yet (as you can see) the dice in column three changes occasionally, in column two more so, and all the time in column one. Thus, we create $1/f$ random numbers. These random numbers are very closely correlated due to the always changing one's column, and the least changing two and three columns. Take a look at the flow chart in **Figure 2**.

About the Arduino Program

I've put in a lot of comments in the code (that's available at the article link), so it should be fairly easy to





■ **FIGURE 3.**
Arduino
schematic.

For the random tones, it really only requires four parts:

- Small Speaker
- 1K Resistor
- BC547 NPN Transistor
- 100 ohm Resistor

Most electronic hobbyists will have these things laying around. The schematic (**Figure 3**) has all the values for each part.

all, I've been programming a long time, and the variables in my 72 year old brain are slowly disappearing, so keep it simple; we have a lot of young brains to teach.

Next, we come to the notes in a sequence section. In this section are a bunch of notes that we can use to play with. First, we have notes from C₃ to A₄ which is about midrange on a piano:

```
int sequence[] =
{ 1, 1, 1,  NC3, ND3, NE3, NF3, NG3, NA3, NB3, NC4,
  ND4, NE4, NF4, NG4, NA4,  1, 1, 1 };
```

This is the sequence we use first. It allows you to hear what 1/f music is like. Then, you can try different scales or create different note sequences for yourself.

The sequence is an array. The three 1s at the beginning are there because when we add up three dice with ones being thrown, we come up with the number 3. When you access an array, it always starts at 0, so we have to start our notes at the fourth position, or 3. The 1s at the end are for rests or no notes played. 1/f notes will have a tendency to hang around the middle of the sequence, so rests aren't often played; after all, you don't want to hear too many nothings.

As you go on with trying out other sequences, comment the first sequence and then uncomment the sequence you want to try. For instance, notes in the pentatonic scale are oriental in nature.

If you want to go farther in trying out different programming techniques — such as making the array smaller — then put the size of the array in the random function. We can determine the length of the sequence array, *lenSeq*, by using the Arduino function *sizeof()*. *Sizeof*

follow what I did.

However, let's go to the program to see how it does all this. Load the Arduino sketch *One_Over_F_Random_Numbers.ino* or view the *One_Over_F_Random_Numbers.txt* file at the article link.

As in all Arduino programs, we have various sections. The first section in this one describes the circuit diagram. The Arduino circuit diagram can be seen in **Figure 3**. I chose an Arduino Uno, but almost any Arduino can be used. By changing the resistor in the speaker circuit (R2), the volume can be changed a bit.

All the notes are listed in the *note_table.h* file. It's added to the Arduino program by clicking Sketch, Add File. The note values can then be compiled into the program by the *#include "note_table.h"* line. If you look at the note table, the notes are labeled like NA1 for example. The N was added because A1 is an Arduino word representing the analog pin 1 and will produce a syntax error. All notes, therefore, are prepended with an N.

The next section sets up the variables in the program. In order to determine the lengths of notes, we have to decide on what tempo we want the notes to play at. The tempo is one minute divided by a number; in this case, 120 beats per minute. This gives us the length of a quarter note (Q in the program). From there, we can use simple math to determine the lengths of the other common notes: sixteenth (S), eighth (E), and whole (W). If you want to play things slower or faster, this is the number to change.

We are generating a lot of random numbers, so they are put into variables *Rnd0* to *Rnd7*. I know that we can avoid a lot of the variables through smart programming, but I have a tendency to keep things understandable. After

RESOURCES

Wikipedia link to the KIM-1 computer
<https://en.wikipedia.org/wiki/KIM-1>

Wikipedia link to the TRS-80 computer
<https://en.wikipedia.org/wiki/TRS-80>

Wikipedia link to the IBM Selectric typewriter
https://en.wikipedia.org/wiki/IBM_Selectric_typewriter

Link to the Scientific American website
www.scientificamerican.com/magazine/sa

returns the number of bytes in the array. Be careful of this length, however. It could be counting more than just the data in the array. Test this very carefully.

The length of the notes – or note durations – is found in another array: *noteDurations*. Since we are 1/f randomizing the durations, we are only using two dice instead of three. The beginning will therefore start at 2 in the array.

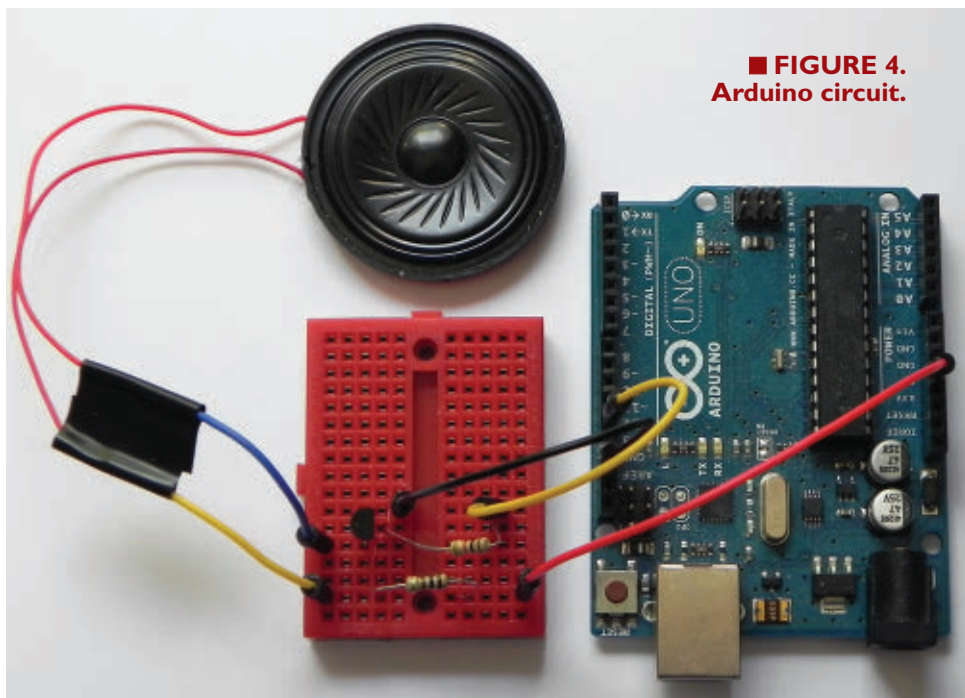
The standard Arduino setup section is next. We make sure the pin the speaker is on is set as an output, and we set the serial port so we can see the note values (and other values) if we need to.

Next, we set a random seed from an unset analog pin. Check out the Arduino reference on this online if you need to see why we do this. Basically, it's to make sure the random numbers created are as random as possible.

Finally, we get to the main loop of the program. We use a counter to create the random numbers from 0 to 7 from the *NumColsToChangeN* array. Remember, its numbers are 3, 1, 2, 1, 3, 1, 2, 1:

```
// Play 8 random notes
for (i=0; i<8; i++) {
  Row = NumColsToChangeN[i];
  switch (Row){
    case 3: // Throw 3 dice
      Rnd1 = random(1, 7);
      // 1 to 6 dots
      Rnd2 = random(1, 7);
      Rnd3 = random(1, 7);
    case 2: // Throw 2 dice
      Rnd1 = random(1, 7);
      Rnd2 = random(1, 7);
    case 1: // Throw 1 di
      Rnd1 = random(1, 7);
  }
  Rnd0 = Rnd1 + Rnd2 + Rnd3;
  // Min 3 (1+1+1), Max 18
  (6+6+6)
```

So, the first number will be 3, and we get three random numbers. The *random(min, max)* function has a silly way of creating numbers. *Min* is inclusive and *max* is exclusive; this means if you want four random



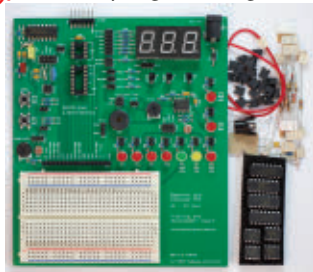
■ FIGURE 4.
Arduino circuit.

numbers, *min* will designate the lowest number (say, 1) and *max* must be 5. Maybe someday this will be fixed.

The first time through the loop we are throwing three dice. We get a random number from one to six that represents the dots on a dice. Next, we get the second and third random numbers. We add them up, and the sum gets stored in *Rnd0*. Then, we get the note length in a similar manner and that gets stored in *Rnd6*. Finally, we get just a regular random rest length:

Premium tools for makers and creators

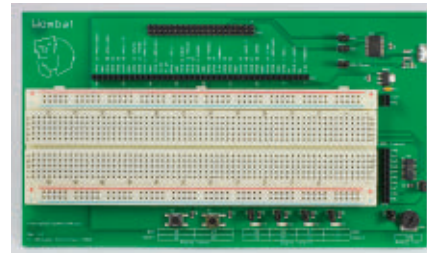
Learn PIC programming



Tutorials and training board

Clear lessons
C and assembler
Hands on examples
Includes enhanced mid-range PICs

Create original projects for Raspberry Pi



The Wombat prototyping board

Provides easy access to all GPIO pins
Adds functionality to your Pi, such as:
analog inputs, LEDs, pushbuttons,
USB serial console port
Includes a set of projects to get you started.



See our Joey - a 7-segment LED display for Raspberry Pi!

www.gooligum.com

```
// Get a rest length
Rnd7 = random(0, 3);
// Get a random # from 0 to 2 (S,E,Q)
rest = restDurations[Rnd7];
```

This could be 1/f generated if you want to do that, as well. The actual rest length (in milliseconds) comes from the *restDurations* array, but we are only looking for a sixteenth, eighth, or quarter note in this program. You can add in more notes if you want. The note duration is pulled from the *noteDurations* array, and the note frequency is pulled from the sequence array:

```
// Play the note or rest
duration = noteDurations[Rnd6];
freq = sequence[Rnd0];
// This plays a rest
if (freq == 1) {
    playRest(duration,rest);
}
if (freq > 1) {
    playTone(duration,freq);
}
```

```
//ShowIt(); // For testing
```

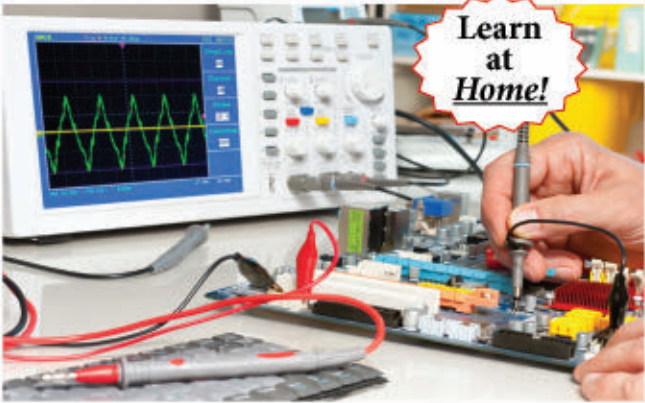
If the frequency is a 1, we play the rest with the *playRest* function. If the frequency is more than a 1, we play the note with the *playTone* function. Finally, if we want to see any variable, we run the *ShowIt* function.

I pulled the *playTone* function off the Internet (thanks to those who wrote it); basically, you provide the duration (*duration*) and the frequency (*freq*) to it, and it pulses the speaker pin. The *playRest* function doesn't pulse anything; it just provides empty time.

After eight notes, we start the whole thing over again. You can throw in delays in the loop and the *ShowIt* function to slow things down a bit, which is probably needed when you want to see the variables in the serial monitor on the screen. You can also make the length as long as you want by changing the index.

So, that's that. Play the different sequences and see how they sound. Do they actually sound like something resembling music? I think they are more musical than just plain random numbers. Even my musical professional wife thought so.

There are MP3 files at the article link for you to listen to. One is a sequence from the middle notes of a piano;



Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.


FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- Industrial Electronics with PLC
- Electronics with FCC
- Electronics Troubleshooting

- **NEW!** Computer Security Specialist
- Broadcast Engineering
- PC Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H





AP CIRCUITS

PCB Fabrication Since 1984

As low as...

\$9.95

each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com







the next one is a sequence from a Pentatonic scale which (as mentioned previously) is oriental in nature; and the third one is a sequence from the notes in the movie, *Close Encounters of the Third Kind*.

Go to www.scientificamerican.com/magazine/sa and buy the magazine (April 1978, Vol 238, Issue 4); it only costs \$7.99. The 1/f random numbers article is in Martin Gardner's Mathematical Games section titled, "White and brown music, fractal curves, and one-over-f fluctuations."

Maybe there's a better program that can be created to make better computer music. Are you one of those who can write it? **NV**



11th ANNUAL **Maker Faire**
BAY AREA
GREATEST Show & TELL ON Earth

PRESENTED BY 

MAY 20-22 SAN MATEO EVENT CENTER
7TH ANNUAL WORLD MAKER FAIRE NEW YORK
OCT 1+2 NEW YORK HALL OF SCIENCE

GET YOUR **TICKETS** TODAY!

makerfaire.com BROUGHT TO YOU BY 



MAKE YOUR MACHINE **MOVE**

MICRO LINEAR ACTUATORS for RC devices

specifications 10mm-200mm stroke
6v-12v power supply
25kg+ available force
15g-100g net weight



 www.firgelli.com

The Personal Computer in Amateur Radio

By John Thompson

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/May2016_PCs_Amateur_Radio.

Since most *Nuts & Volts* readers are already computer/kit/Raspberry Pi/Arduino users, it will warm your heart to learn of computer uses in amateur radio. What? Amateur radio? That bunch of retro geeks? Yes! The initial use of computers in ham radio involved logging contacts with other ham stations — especially for use in ham radio contests where the score is the number of contacts times the multiplier. An extensive list of contests is available at wa7bnm.com.

This discussion will be primarily about ham radio contesting programs, with an additional overview of all the myriad uses of computers in ham radio. The initial popular programs were by WJ2O and K1EA. K1EA's CT revolutionized ham radio contesting in the DOS days. Back then, you needed different programs for different contests, including TR by N6TR and NA.

The initial ham radio logging programs (for DOS) were hindered by the necessity of pre-loading port drivers for the DB-9 RS-232 output ports. Modern programs do not require this. In fact, almost all can be run with a USB type of output, without the need for the cumbersome DB9-COM port converters, which suffer from almost a universal lack of standardization.

When Windows arrived, so did WriteLog (WL). This

program — originally written for radioteletype (RTTY) — was far advanced as compared to its predecessors. Then, the free programs N1MM and M1MMPlus took over. Alternatives also include Win-Test and SuperDuper. It took me two years to learn WL and seven years to use N1MM, but N1MMPlus is by far the most favorite current contest program among the general ham radio contesting population. More computer-savvy individuals know how to use this program without any instruction.

A screenshot of N1MMPlus is shown in **Figure 1**. Free instructional videos are on the WWROF foundation website. N1MMPlus is free, and was written by a dedicated group of about 10 hams. All modern logging programs have a feature that can indicate the frequency of operation of the transceiver and that a telnet spotting

network is accessed, which together markedly increase scores. A DX spotting network is one that shows the call signs and frequency of stations that are desirable with which to make contact for fun, or an increased score in the contest.

For CW (international Morse code) and RTTY contests, the reverse beacon network greatly increases scores over telnet nodes. This is a system of automatic receivers/decoders that spot stations (rather than humans) all over the world that are being received. It is free to users. The RBN (Republic Broadcasting Network) system is revolutionizing amateur radio contesting to the same extent that the invention of the logging program did in the '90s.

Besides keeping score and sending (sometimes receiving) code and voice, "ham" logging programs have a "supercheck partial" system where — in the case of an incompletely copied call — the program suggests a call that might fit in from a database of

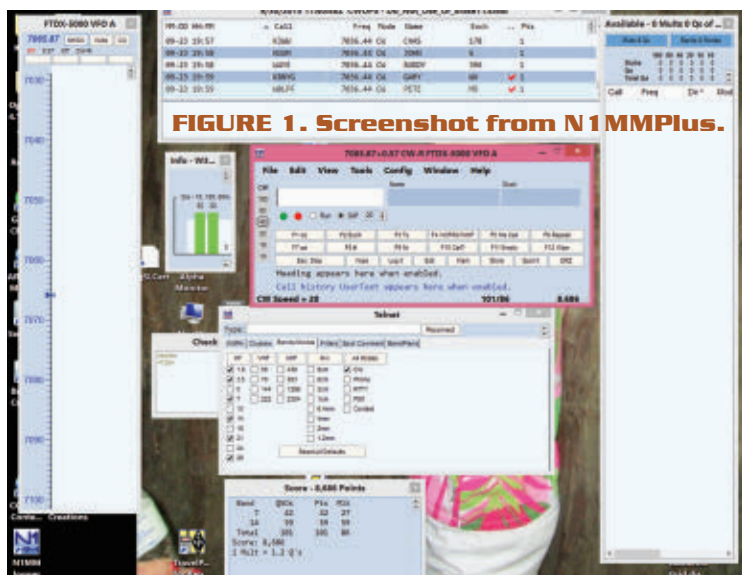


FIGURE 1. Screenshot from N1MMPlus.

active hams that is updated monthly. This is extremely useful in the instance of poor band conditions. Also, the program keeps track of the solar flux and K index which predict solar storms and good or bad communications on high frequencies.

Common free programs such as MMTTY and Digipan allow any ham with a PC to easily transmit and receive radioteletype and phase shift keying transmissions. For an “idiot proof” type of installation (recommended), just buy the West Mountain Radio Soundblaster Plus USB set up with the cable specifically for your radio; then plug and play. The software is included with the adapter unit. Totally easy! There are MUCH more expensive units available (microHam), but for a basic unit, West Mountain Radio cannot be beat. Homebrew systems can also be made. Most laptops can be configured for digital communications with little or no modification. You might need a USB/COM port adapter. There is little or no standardization in USB/COM port adapters, so use the one the interface manufacturer suggests.

The modern HF ham radio transceiver is in itself a computer. First of all, there is a phase locked loop or direct digital synthesizer with a digital frequency readout. Yaesu first came out with the FT-1000MP transceiver around '95, which offered digital signal processing in the audio chain. This allowed for a marked reduction in broad spectrum (lightning, ignition) noise interference, and very good “auto-notching” of carrier signals that were interfering on single sideband. In the current era, a transceiver (transmitter-receiver) for \$750 (FT-450; see www.aesham.com) will feature digital signal processing in the intermediate frequencies, which is far superior to that done at audio frequencies. This is far removed from the tuned circuits only of radios up to 1995.

For the complete and true computer geek, the system is the flex series of transceivers. This system uses a very fast PC to develop a signal with digital signal processing and a direct conversion type of system. These early units required the use of a firewire connection between the unit and the computer; more recent ones have the computer built in. However, bring your pocketbook. These are some of the priciest radios per feature on the market. They make fancy used test equipment look cheap by comparison.

There is an elaborate (but simple to use)

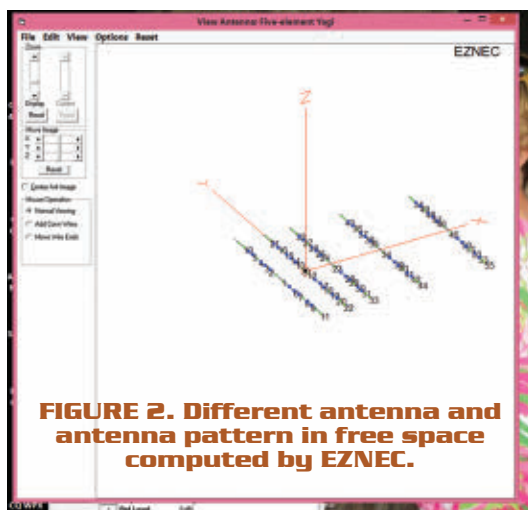


FIGURE 2. Different antenna and antenna pattern in free space computed by EZNEC.

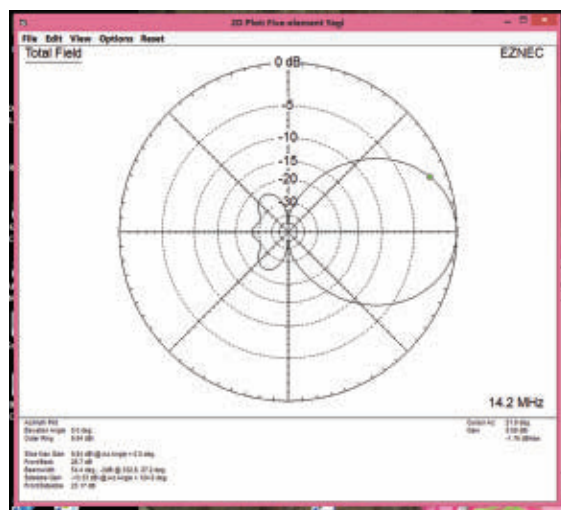
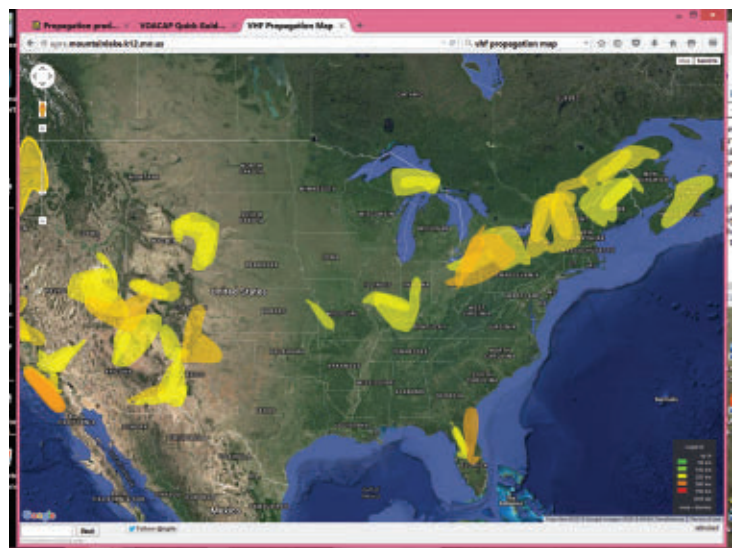
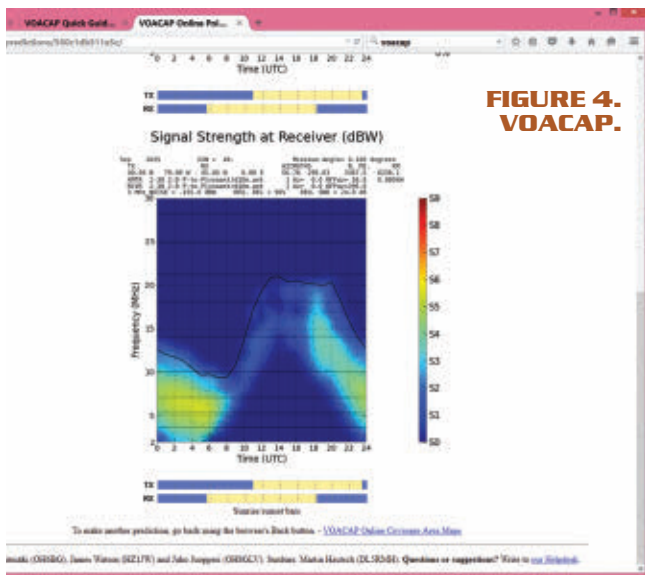


FIGURE 3. Antenna and antenna pattern in free space computed by EZNEC.

program for modeling antennas written by W7EL. A simplified version of this is available free with the *ARRL Handbook*. With this program, you separate an antenna into a series of “wires,” specify the diameter of the material (tapering is allowed), and then you can optimize the antenna in free space or above an ideal ground or real ground. An example of a five element Yagi type antenna (as in TV type antenna) is shown in **Figures 2 and 3**.

Predicting when and where to listen and transmit for a given communications path has been studied since the days on Marconi. In the “good old days,” there were endless articles in *CQ Magazine* on propagation prediction. Now, it is a simple matter to download a program and plug in the numbers (solar flux, A and K indices), then the program will plot a signal strength over time with a given power output, antenna, and transmitter/receiver frequency. Most high frequency skip is

Application	Resource
Antenna modeling	www.eznec.com
Contest logging	www.n2mmplus.com
	www.writelog.com
	www.wa7bnm.com
	www.wwrof.org
Digital	www.dxatlas.com/CwSkimmer
	http://hamsoft.ca/pages/mmtty.php
	www.digipan.net
	www.w1hkj.com/
	www.hamgadgets.com/ULTRA-PICOKEYER
General radio logging	www.n3fjp.com
	www.dxlab.com
	www.dx4win.com
Moonbounce/meteor scatter	www.pingjockey.net/cgi-bin/pingtalk
	physics.princeton.edu/pulsar/k1jt
	www.qsl.net/kd2bd/predict.html
Propagation	www.dxzone.com/dx5020/w6elprep.html
	www.voacap.com
Transceiver as computer	www.icomamerica.com/en/amateur
	www.yaesu.com
	www.flexradio.com/amateur-products
Schematic drawing	www.digikey.com/schemeit/#
Satellite passes	www.amsat.org/amsat-new/tools/predict



from the F2 layer. The most popular programs are VOACAP and K6ELPROP. **Figure 4** is an example of VOACAP. As retro as it seems, the use of CW communication is on the rise. The primary reason for this is that it is more reliable in an emergency situation. These days, it is a simple matter to find a program that will print out code such as Writelog, Fldigi, or Skimmer. Morse code communication is more reliable when signals are weak or interfered with by atmospheric noise by a figure of at least

The Easiest Way to Design Custom
Front Panels & Enclosures

Free Front Panel Designer

You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

FRONT PANEL EXPRESS
FrontPanelExpress.com

**WORLD'S MOST VERSATILE
CIRCUIT
BOARD HOLDERS**

Model 324 Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

Model 201

MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

PANA VISE®
Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7535 | www.PanaVise.com

3 dB (two times). With the inverse square law, this will yield communications at 1.44 times the distance.

For repetitive exchanges, the transmitter can be keyed from the program, or from a "contest memory keyer." By far, the best buy in this area is the ultra Picokeyer kit for \$29.95, which has features that used to cost \$150 in 1980. Any kit builder can successfully construct it (if I can build it, *anyone* can build it). While you are at it, you can build up a digital voice keyer (DVK) for \$10 per module — not a bad price. This will record your voice and send it on command, while keying the transmitter also.

No discussion of computer use in amateur radio would be complete without mentioning antenna positioning software. This can be useful for earth-moon-earth (EME) communication or for amateur radio satellite communication. EME is done with high powered transmitters and very sensitive receivers with directional antennas, often having a gain of over 20 dB (100 times). Modern EME communication is done with time-domain averaged software which was developed by K1JT at Princeton. This required only 200 watts of power output to a single Yagi (14 dBd gain). There is also a high speed frequency shift keying technique for bouncing radio signals off of meteor trails (you don't have to belong to NASA to do this ... just become a ham operator).

Personally, I avoid the use of antenna positioning software, as the contest program antenna positioning changes are very likely to break your rotator (rotor) from overuse. The bigger your array, the more likely it will windmill in your rotor mount or simple destroy the rotor. I have destroyed three rotators in the past 12 years, and I don't even have a big array (**Figure 5**). Direct connection between the PC and the rotor (Yaesu, Green Heron Engineering) is available.

One of the splinter areas of ham radio is that which involves the use of high power transmitters, sensitive receivers, and high gain antennas on very high frequencies and on microwave. On six meters (50 MHz), skip of around 1,200 miles per hop is available during the summer. On higher frequencies, the wave just penetrates the ionosphere, but it is refracted back towards the earth in some cases. There is a wonderful website which continuously updates the ionospheric refraction of VHF

signals shown in **Figure 6**.

There is a subset of folks who use the moon and meteor scatter in order to make contacts. In the 1980s, this required the use of a mast mounted preamp, homebrew antenna changeover, 20 dB antenna system (100 pounds of antenna), and 1,000 watts at the antenna. As stated earlier, it now takes only 200 watts and a single ham radio Yagi gain antenna.

So, if you want to combine digital communications with the ability to talk internationally (with or without the Internet or Wi-Fi), get you ham radio license! How? Study from **qrz.com** and find a local ham radio club; they give the exams. **NV**

Get the Parts You Need FAST!

**HIGH QUALITY ELECTRONIC COMPONENTS
IN STOCK AND READY TO SHIP**

nte parts direct®
Quality is Always In-Stock

- Consumer, commercial, industrial and MRO applications
- Huge product inventory
- Wide selection and complete specs
- No minimum order
- Cross referencing for thousands of parts
- Same day shipping available

ORDER TODAY!

www.ntepartsdirect.com

OLD HEATHKITS NEVER DIE, THEY JUST GET RESTORED

I love restoring old electronic equipment, bringing things back to life. A while ago, I was excited to find a vintage Heathkit EC-1 analog computer for a reasonable price on eBay. It definitely needed a little TLC.

The Heath Company originally sold the EC-1 from 1959 to 1971. It had 17 vacuum tubes and lots of knobs and banana jacks in an awesome front panel. **Figure 1** is a picture from the Heathkit catalog. Nowadays, they show up on eBay on a fairly regular basis, but some of them can be quite expensive.

NOTHING IS EVER SIMPLE

When I bought the computer, I knew I was in for a major cleanup after seeing the photos on eBay. **Figure 2** shows the unit just after I unpacked it. At first, I thought that a little 409 cleaner would do the job, but I soon

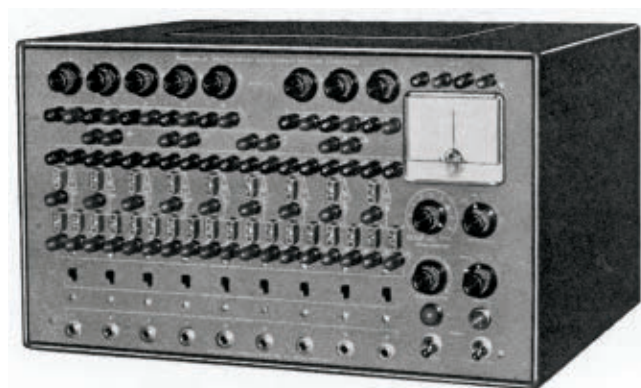


FIGURE 1. The Heath Company sold the EC-1 analog computer from 1959 to 1971 for \$199.95. Today, they can sell for well over \$1,000 on eBay.

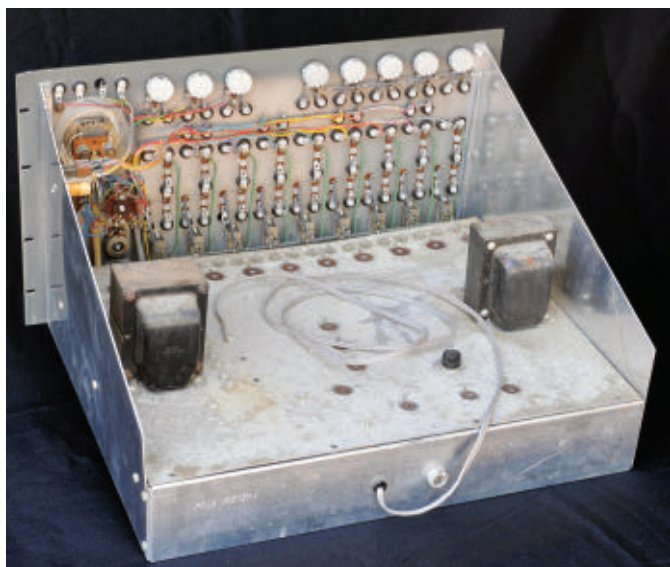


FIGURE 2. This EC-1 was purchased on eBay for \$420. It obviously needed a major restoration.



FIGURE 3. The finished restoration included new vacuum tubes, tube sockets, binding posts, potentiometers, and a line cord.

realized that I would have to completely disassemble it and unsolder every component in order to bring it up to my standards.

I wasn't planning on restoring it to an all-original museum piece, but I did want it to look nice and work. The finished restoration is shown in **Figure 3**. I was pleased with the result. The old metal cabinet had some light rust on it and I had planned on trying to match the original Heathkit gray wrinkle paint. After unsuccessfully trying a few automotive sprays, I went for the ultimate: a gray hammertone powder coat. The finish looks great (as you can see in **Figure 4**) and it's tough as nails.

The front panel was missing a red binding post, and the rest of the posts were dirty and tarnished inside. I tried finding a matching post, but in the end I bought six dozen brand new ones. Their bright color gives the front panel a real pop.

Taking the unit apart was easy. It was putting it back together and pre-testing each circuit that took the time. I took lots of "before" digital pictures and made sketches of the critical wiring so I wouldn't lose track of things. A few days later when I was surfing the Internet, I was amazed to come across scans of both the original Heathkit EC-1 assembly and operational manuals. They were available as two 40+ page pdf files and contained complete schematics for the amplifiers, power supplies, and front panel controls. To download the manuals, please see the web address in the **Resources Section**.

SOME ADVICE

If you are ever thinking about buying and restoring a piece of electronic equipment, I have several bits of advice for you to consider:

1. Be sure the front panel and its silk screening are as clean as possible. It can be difficult to restore the delicate silk screening or big scratches. Plus, any calibration or inventory stickers can leave discolored areas.

2. If you are buying a unit on eBay, be sure you see at least



FIGURE 4. The new binding posts enhanced the look of the front panel. The nine "amplifier balance" pots along the bottom had degraded during the past 50 years and were replaced.

one photo of the interior. Sometimes bad things can be found within, like overheated transformers, water damage, missing parts, etc.

3. A little grime is okay, but extensive corrosion or a broken one-of-a-kind component can be hard to replace or restore.

GETTING TO WORK

One of the big no-no's with vintage equipment is to plug it in the wall to "see if it works." Even using a variac is not the best way. Some components — like electrolytics — tend to degrade with time, and if you apply full voltage they can actually explode. So, the trick is to either "re-form," rebuild, or replace them. The Internet contains a plethora of excellent advice on how to re-form electrolytics. Sometimes it's successful, sometimes not. For this restoration, I chose to rebuild the electrolytics and replace the smaller caps with modern ones.

Figure 5 illustrates the four steps I took to dissect the old electrolytics and slide the new ones inside the paper sleeves. I thought that using the original sleeves would give the restoration a more vintage look.

After all the other components had been unsoldered, I measured a number of the carbon resistors and potentiometers (pots), and was surprised to find that many were out of spec. So, I decided to replace them all. The wire-wound power resistors were fine.

The only major deviation from using all original parts was the replacement of the nine "amplifier balance" pots

located along the very bottom of the front panel, as seen in **Figure 4**. During the past 50+ years, the original pots had degraded in value, from 3,000 ohms to as low as 2,150 ohms, and several of them didn't even have enough range to balance the amplifiers. The original pots had screwdriver adjustment slots, but I chose to buy pots with shafts and add a small knob because this way, they were much easier to adjust. They made the front panel look different from the original, but what the heck.

The rectifiers were a strange mix of silicon and series-wired

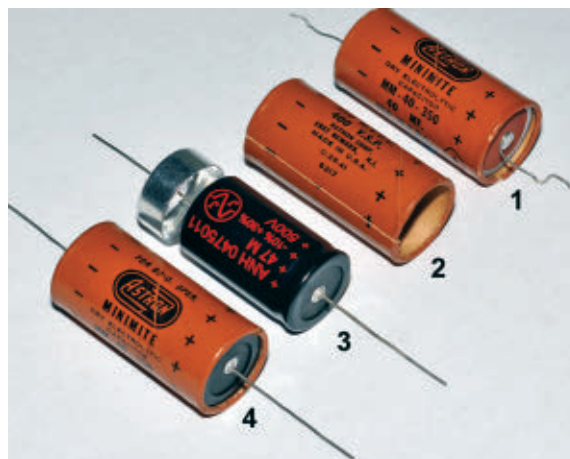


FIGURE 5. Rebuilding the large electrolytic capacitors involved slipping new capacitors into the old paper sleeves to retain the vintage look.



FIGURE 6. The chassis was completely rewired using a mix of original and new parts. The five parallel bus bars distribute the +300 VDC, -150 VDC, and 6.3 VAC.

selenium diodes. I had never seen a potted selenium rectifier, so I cracked one open and found the usual stack of plates; they were small, rated at 50 ma. A photo of them can be found in the files at the article link as **Figure A**. In days past, if you've ever experienced an overheated selenium rectifier, you will never forget the unique smell. In this restoration, they all were okay, so I remounted them in the chassis. I replaced the dissected unit with a modern silicon diode.

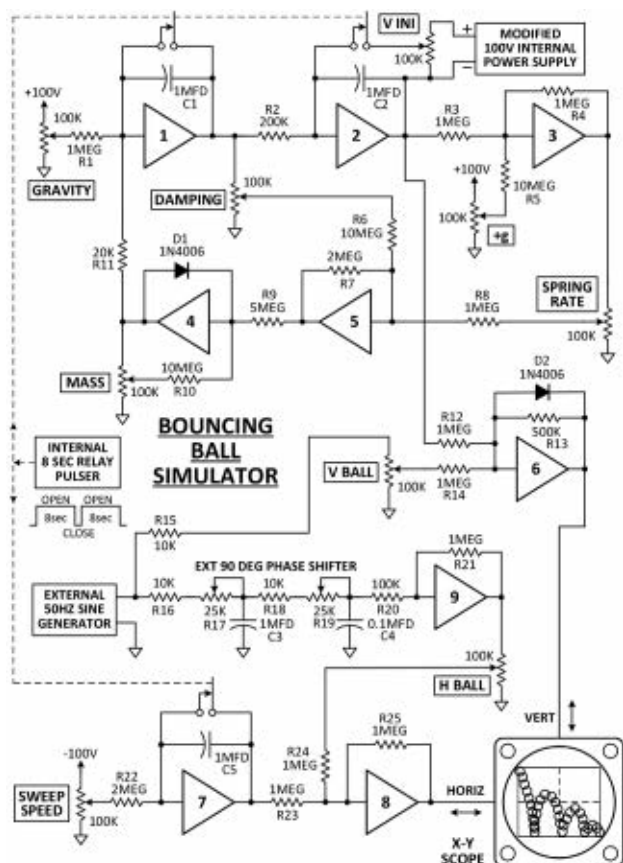


FIGURE 7. One of the classic demos for an analog computer was to simulate a bouncing ball. The op-amps were configured as summers, multipliers, dividers, integrators, and inverters.

The Rise and Fall of Electronic Analog Computers

In 1949, the large electronic analog computing machine shown in the figure was located at the Lewis Flight Propulsion Laboratory in Ohio. It contained thousands of vacuum tubes and was capable of modeling fluid flow, pressure, temperature, and other parameters required by the early space program. The advantage of this type of computer was that it could electrically model and simulate many fluidic and mechanical systems without having to actually build and test expensive hardware. They were used to run varying simulations of a system and when the results had converged on an optimal solution, the actual hardware could be designed and constructed with a greater confidence that it would meet the required performance.



Several of the larger analog machines included Enrico Fermi's FERMIAC, Reeves' REAC, Beckman's EASE, and Project Typhoon developed by RCA in 1952. The RCA machine contained over 4,000 vacuum tubes, 100 dials, and 6,000 plug-in connectors. The simulations were set up by plugging numerous patch cords into large patch panels.

However, as digital computers became more capable, the hybrid digital/analog computer was born and stand-alone analog computers began to fade. Eventually, digital computers had advanced enough in speed, memory, and software to handle models and simulations that exceeded even the capabilities of hybrid computers, so they became obsolete. Fortunately, some of the original analog machines can still be found in museums around the world for visitors to enjoy.

BUYING NEW PARTS IS FUN

On the upside, the most enjoyable thing for me was ordering new parts from places such as Antique Electronic Supply (www.tubesandmore.com) in Tempe, AZ. They have hard-to-find things like high voltage capacitors, wafer-type tube sockets, square bus bars, and terminal lug strips.

The computer arrived at my door with only half its vacuum tubes, but that was a trivial problem because many types of tubes are still available and not too expensive. Fortunately, the EC-1 didn't use any of the outrageously expensive power tubes used in vintage guitar amplifiers.

My biggest worry was breaking one of the delicate contacts on the wafer switches. The switches would be almost impossible to replace because of their custom wipers. So, I wicked the solder off the lugs, carefully removed the wires, and luckily nothing broke. The wiping contacts were tarnished a bit but a little silver cleaner worked wonders.

The aluminum chassis had some minor corrosion, so I used a small handheld orbital sander to moderate the spots and to impart a more brushed look to all the chassis panels. It helped to hide the minor scratches and abrasion marks. The final prep was a scrubbing with the type of green abrasive pad typically found in kitchens, which gave the panels a nice luster.

FINAL ASSEMBLY AND TEST

My approach to the final assembly involved wiring up and testing the power supplies first. I love OA2 and OB2 regulator tubes with their warm glow discharges. Once the +300V, -150V, and three 100V power supplies were checked out, the rest of the components easily fell into place using the assembly manual, digital photos, and notes taken during disassembly. It's always sweet to replace old brittle rubber grommets with new pliable ones. **Figure 6** shows the new and rebuilt parts mounted in the chassis. Notice the five parallel bus bars which distribute the AC and DC voltages to the operational amplifiers (op-amps).

Actually, firing up the reborn Heathkit was a non-event since I had thoroughly tested or replaced almost every component. The nine op-amps balanced fine using the new pots, and adjusting them with their new knobs was much easier than messing around with a screwdriver.

It was then that I hit the proverbial wall. Now that everything's working, what's next? My friends thought it looked nice but they wondered what it was good for. Then, I remembered a diagram in the back of the EC-1 operational manual that described an interesting application: the simulation of a bouncing ball. The potentiometers on the front panel could be set up to vary the effects of gravity, mass, damping, and spring rate on the ball. The output signals would go to an X-Y oscilloscope so you could graphically see the motion of the ball.

BOUNCING BALL SIMULATION

Figure 7 is the schematic of the simulation, which used all nine op-amps, dozens of resistors, a few capacitors, and a couple of diodes. Each op-amp was basically an inverting amplifier with an open loop gain of approx. 1,000 and an output range of $\pm 60V$. FYI, a schematic of the nine identical op-amps is available at the article link as **Figure B**. A listing of the components used for the simulation can be found in the **Parts List**.

The operational manual that I downloaded had a large section devoted to explaining the basics of using op-amps to do addition, subtraction, inversion, multiplication,

integration, and differentiation. Plus, it thoroughly explained how the op-amps were used to implement the differential equations used by the bouncing ball simulation.

However, I realized there was something missing. The original kit of parts had included 27 two-pin clear plastic plugs to hold the simulation resistors and capacitors. In turn, the plugs would have been plugged into the 27 two-pin brown sockets on the front panel, as seen back in **Figure 4**. The unit I purchased had no plastic plugs! I checked eBay ... no joy. Of course, I could have simply slipped the resistor and capacitor leads into the holes in the red and black binding posts and then tightened down the knurls, but that would have been much too easy. I wanted the original plugs or something like them.

The answer was to use my Grizzly benchtop milling machine to make them. I changed the design a bit and potted the resistors in hollowed out blocks of delrin plastic. The pins I needed just happened to be the same size as the pins in old octal vacuum tubes, so I yanked a bunch of them out of 6SN7s and 6J6s. Then, I crimped them onto the component leads and potted the assemblies into the plastic blocks. Done. A photo of the potted resistors and diodes can be found at the article link as **Figure C**.

A RAT'S NEST DID THE JOB

Using the simulation schematic, I crisscrossed the front panel with 35 red and black Pomona patch cords, from 4" to 24" long; refer to **Figure 8**. It was a real rat's nest. I couldn't stop myself from displaying the output on a vintage Heathkit IO-10 3" scope that I had recently restored. I had other scopes, but I thought the little IO-10 was a fitting match. I've always liked the IO-10 because it had knobs that were somewhat similar to the ones found on Tektronix scopes.

Figure 9 was the grand finale. The simulation cycle

Parts List		
Designator	Component	Digi-Key, unless noted
R1, R3, R4, R8, R12, R14, R21, R23-R25	Resistor, 1 meg, 1/2W	1.0MH-ND
R2	Resistor, 200K, 1/2W	200KH-ND
R5, R6, R10	Resistor, 10 megs, 1/2W	10MH-ND
R7, R22	Resistor, 2 megs, 1/2W	2.0MH-ND
R9	Resistor, 5 megs, 1/2W	5.0MH-ND
R11	Resistor, 20K, 1/2W	20KH-ND
R13	Resistor, 500K, 1/2W	510KH-ND
R15, R16, R18	Resistor, 10K, 1/2W	10KH-ND
R20	Resistor, 100K, 1/2W	100KH-ND
R17, R19	Potentiometer, 25K	*R-V38-25KL
C1, C2, C3, C5	Capacitor, 1 mfd, 630V	*C-FS-630-1MFD
C4	Capacitor, 0.1 mfd, 50V	399-4151-ND
D1, D2	Diode, 1N4006	1N4006FSCT-ND
* Antique Electronic Supply		

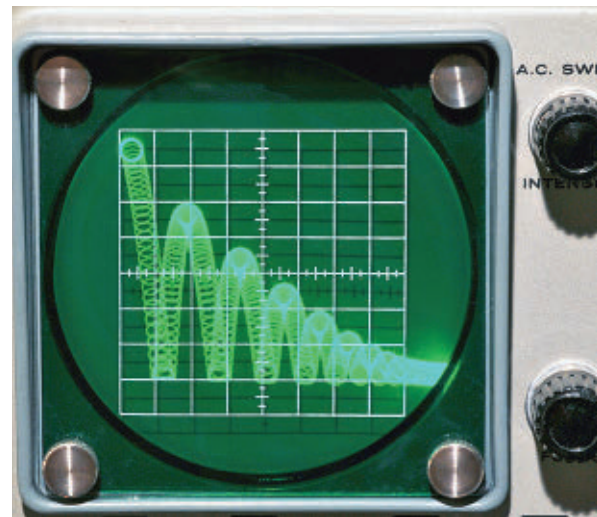


FIGURE 9. An X-Y oscilloscope was used to display a bouncing ball. The repeating horizontal sweep was approx. eight seconds. Notice that the ball flattens out on the first several bounces.

FIGURE 8. The simulation required 35 patch cords, 25 resistors, 11 pots, five capacitors, two diodes, and all nine op-amps.

relay was set at an eight second period, and when the relay contacts opened, the ball dropped and bounced for eight secs, then returned to the initial starting position and repeated. It was mesmerizing. The various effects on the motion of the ball were adjustable by the pots on the front panel over a wide range.

You could simulate the gravity on the moon or the earth, and the simulated ball could be made of steel or even a super ball that never stopped bouncing. There were plenty of variations to explore.

Although looking back, all was not well when I first

turned on the simulation. The ball on the scope was a mess of zig-zag lines — not round at all. A 60 Hz hum was somehow getting into the op-amps. After much hair pulling, I traced the problem to the Vertical Initialization (V INI) power supply. It wasn't isolated enough, so I added a small 120 VAC transformer under the chassis and the ball cleaned up.

In closing, the EC-1 was a fun and challenging project, and I learned a great many new things. However, I think I've taken the bouncing ball as far as it can go, so now I'm casting around for other interesting simulations to try.



WHAT'S NEXT?

In the meantime, I have a new quest: to search eBay each and every day until I find the quintessential Heathkit analog computer — the ES-400. It has even more op-amps and vacuum tubes than the EC-1, plus a sloped panel packed with enough components to take your breath away. Just Google the Heathkit ES-400 and you'll see what I mean. **NV**

RESOURCES

Vintage electronic parts
www.tubesandmore.com

EC-1 manuals
www.computerhistory.org/collections/catalog/102649920

Catalog of Heathkits
Heathkit Test Equipment Products
by Chuck Penson

Vintage Computing

Five More Projects for the Mentor's Friend

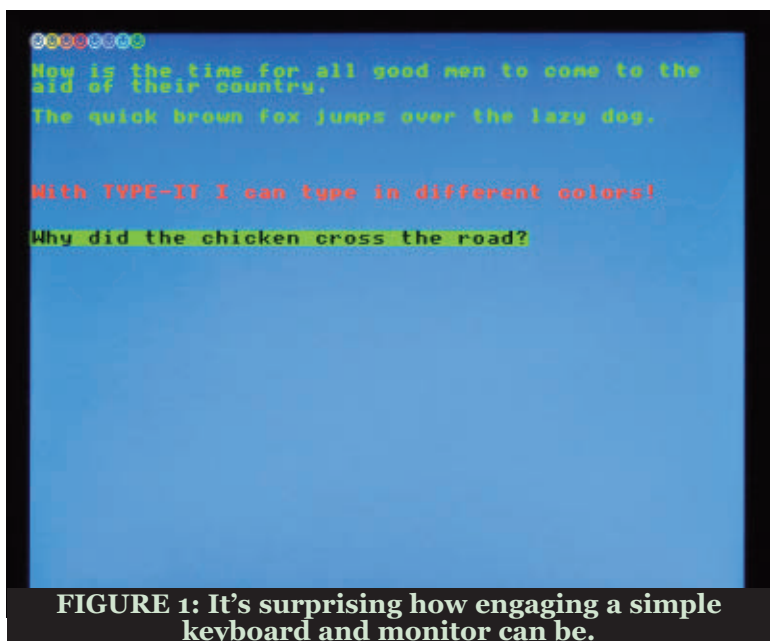
By Dane Weston
daneweston15@gmail.com

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/May2016_RetroComputer_More_Amigo_Games.

Previous issues of Nuts & Volts introduced the Mentor's Friend: a retro computer that you can build yourself and program in BASIC. The "Amigo" was designed as a home-built mentoring platform you and your child or grandchild can use together to explore computers and programming. Here are five more easy Amigo projects for you and a young protégé. These have been selected to continue a hands-on introduction to Color BASIC, while preserving the necessary ingredient of "fun." Enjoy!

1. TYPE-IT.BAS.

One of the first things my grandkids wanted to do with their retro computers was just type text on the screen. This helped their keyboarding skills, and more importantly it got them spending a little "alone time" with their new computer friend. Here's a program that displays keystrokes on the monitor, with the bonus of changing the text colors as needed (**Figure 1**).



Use the Editor (press <F1> at the flashing cursor) to enter the code below. Take your time! This is a “real” program, and there’s a bit to it. If you’re working with a young one, you may ask him or her to help by entering all the lines with a COLOR command, or some similar division of labor to keep them engaged. Take a couple of breaks if needed — don’t let the tedium of code entry destroy the fun of the adventure!

Once you’re done, press <F1> again to return to Color BASIC, then <Esc> to interrupt program flow. SAVE the code as “TYPE-IT.BAS,” then RUN the program through its various features and functions to find and correct any typos. (You’re very special — or very lucky — if you have none!) Don’t forget to SAVE again after your corrections. This is a lot of keyboard work, and you want to protect your investment.

Here’s the code to enter:

```
10 REM *** TYPE-IT.BAS ***
20 REM -- Initialize Things --
30 f=63: b=22: c=22 REM <- White on Blue Text
40 REDEFINE 123,255,129,129,153,153,129,129,255
42 REDEFINE 125,60,66,165,129,165,153,66,60
50 REM -- Print Initial Screen --
60 COLOR f,b
70 GOSUB 1000
80 REM -- Start Main Loop --
90 k=INKEY
100 IF k=0 THEN GOTO 90
105 COLOR f,b
110 REM -- Check for Function Key --
120 IF k<208 OR k>213 THEN GOTO 150
130 GOSUB (k-207)*1000 REM <- Execute Command
140 GOTO 90
150 REM -- Check For Backspace First --
160 IF k=200 THEN DISPLAY 8,32,8: GOTO 90
170 REM -- Print Character --
180 DISPLAY k
190 GOTO 90
200 REM -- End of Main Loop --
999 END
1000 REM -- Print Commands Screen --
1005 GOSUB 2000
1010 LOCATE 5,1: PRINT ">>> TYPE-IT Commands
<<<"
1015 LOCATE 3,3: PRINT "F1 - Show This Screen"
1020 LOCATE 3,5: PRINT "F2 - Clear The Screen"
1025 LOCATE 3,7: PRINT "F3 - Change Text Color
(Arrow Keys, Then Enter)"
1030 LOCATE 3,9: PRINT "F4 - Change Screen
Color"
1035 LOCATE 3,11: PRINT "F5 - Print Smiley"
1040 LOCATE 3,13: PRINT "F6 - Quit Program"
1045 LOCATE 3,17: PRINT ">>> Press Any Key To
Continue..."
1050 IF INKEY =0 THEN GOTO 1050
1055 GOSUB 2000
1095 RETURN
2000 REM -- Clear The Screen --
2005 COLOR c,c
2010 CLS
2015 COLOR f,b
2095 RETURN
3000 REM -- Change Text Color --
3005 DISPLAY 123
3010 u=INKEY : IF u=0 THEN GOTO 3010
3015 IF u=192 AND f>0 THEN f=f-1
3020 IF u=193 AND f<63 THEN f=f+1
3025 IF u=194 AND b<63 THEN b=b+1
```

```
3030 IF u=195 AND b>0 THEN b=b-1
3035 COLOR f,b
3040 DISPLAY 8,32,8
3045 IF u<>13 THEN GOTO 3005
3095 RETURN
4000 REM -- Change Screen Color --
4005 COLOR 63,0
4010 CLS
4015 FOR n=0 TO 63
4020 COLOR n,0
4025 PRINT n,
4030 NEXT n
4035 LOCATE 3,15: INPUT "New Screen Color? ";c
4040 GOSUB 2000
4095 RETURN
5000 REM -- Print Smiley --
5005 DISPLAY 125
5095 RETURN
6000 REM -- Quit Program --
6005 CLS
6010 LOCATE 16,16: PRINT "Quit Program (Y/N)?"
6015 u=INKEY : IF u=0 THEN GOTO 6015
6020 IF u<>"y" AND u<>"Y" THEN GOTO 60
6025 REM <- Reset Redefined Characters -
6030 REDEFINE 123,7,12,12,120,12,12,7,0
6032 REDEFINE 125,112,24,24,15,24,24,112,0
6040 REM <- Reset Screen, Print Prompt, and
End -
6045 COLOR 63,22
6050 CLS
6055 PRINT "READY"
6060 END
6095 RETURN
```

Once everything is working, you should have a little program that presents an instruction menu for function keys F1 through F6. It then allows your “student” to type text in various colors on a screen color of their choice, clearing the screen, changing colors, or printing a smiley as needed. It may seem unexciting at first (no animation, no multi-gesture swiping, etc.), but my grandkids seemed to truly enjoy typing their own little jokes and short stories with a real keyboard and screen.

If you’ve worked through the Color BASIC projects in previous issues of this magazine, most of the code in this program should make sense on your first read-through. However, this project is probably close to some that you will build later on, so a general walk-through of the code may be in order.

The program is divided into modules, delineated with REM statements. It consists of a *main* loop to capture and process keyboard inputs, preceded by some necessary initialization, and followed by a series of subroutines that perform desired program features. Here’s what’s going on in each module:

- Lines 10 - 42 initialize the text foreground (**f**), text background (**b**), and screen (**c**) color variables to white on blue, then REDEFINE the right and left brace keys (ASCII 125 and 123) to a smiley and a purpose-built character for changing **f** and **b**.
- Lines 50 - 70 set the text colors and call the Print Commands Screen subroutine at Lines 1000 - 1095. Like any subroutine in Color BASIC, program flow jumps from

the calling GOSUB (in this case, Line 70) to the specified line number (Line 1000), then from the RETURN (Line 1095) back to the line following the GOSUB (Line 80).

- Lines 80 - 200 comprise the main loop of the program, which converts keystrokes into program commands or characters on the screen. Lines 110 - 140 check for Function Keys 1 (ASCII 208) through 6 (ASCII 213). If these are not found, program execution continues at Line 150. If F1 through F6 is presented, Line 130 sends program flow to the appropriate subroutine. Then, Line 140 jumps to the top of the loop for another keystroke. Note the use of the expression $(k-207)*1000$ in Line 130 to select the proper subroutine. In Color BASIC, both GOTO and GOSUB can use expressions like this, as well as explicit line numbers.

- Lines 150 - 160 provide the program with an important capability: the ability to backspace to correct errors. If the keystroke is a <Backspace> (ASCII 200), Line 160 displays an ASCII 8 (move the cursor back one space), an ASCII 32 (print a space over the current character), and an ASCII 8 (move the cursor back one space again). This is a bit confusing at first, but it makes sense after you think about it. After eliminating the offending character, program flow jumps back to get another keystroke.

- Lines 170 - 190 print the character glyph of any remaining keystroke value, then goes back for another keystroke. This process continues until the program is ended by the F6 command.

- Now, let's take a look at the subroutines. Lines 1000 - 1095 print the commands screen, which lists the active function keys and the function each performs. Note the use of GOSUBs in Lines 1005 and 1055 to avoid duplicating code. (Subroutines can "nest" one inside another five deep in Color BASIC.) The strange numbering on Line 1095 and the other RETURN statements are there because I first put a program "stub" in for each subroutine when developing my code — just the REM statement label and the ending RETURN statement. This lets me set up and check my program framework first, and then build (and improve) the functional modules after program flow has been validated.

- Lines 2000 - 2095 clear the screen in the screen color **c**, then set the text colors back to **f** and **b**.

- Lines 3000 - 3095 display a change-text-colors indicator (a rectangle containing a dot) in the current colors, then allow the user to change colors with the arrow keys, pressing <Enter> to return to typing mode. The left and right arrows (ASCII 192 and 193) decrement or increment **f**; up and down (ASCII 194 and 195) increment or decrement **b**. After setting the text colors to the new value and backspacing over the change-color indicator, program flow loops to the top of the routine unless the <Enter> key (ASCII 13) has been pressed. This design has less user prompting than I like, but doesn't require us to

separately track the cursor in our code.

- Lines 4000 - 4095 erase the current screen, present the color value set in the assigned colors, and prompt the user to select a new color. Note that in the call to the clear-the-screen subroutine in Line 4040, all variables are shared among subroutines in Color BASIC. So, the value of **c** set in Line 4035 is available to Line 2005 without any explicit passing of variables.

- Lines 5000 - 5095 simply print a smiley at the current cursor location, in the current foreground and background text colors.

- Lines 6000 - 6095 end the program, first getting user confirmation, then resetting the redefined characters and screen color, then clearing the screen and printing a READY prompt. The RETURN in Line 6095 is never executed, but left in there for consistency.

You should find it fairly straightforward to add features to this code that you and your protégé come up with — more emoticon function keys is a good place to start. If you haven't done so yet, check out the REDEFINE.BAS program that ships with the Amigo kit (available through the *Nuts & Volts* webstore). It should help with developing REDEFINE statements for any new glyphs you implement.



DO MORE THAN PLUG AND PLAY
Build, program, and fly with the ELEV-8 v3 quadcopter.

Engineered in the USA to give you the freedom to learn and experiment. The robust design is manageable to build, customize, and maintain - no soldering required.

Available accessories:

- telemetry kit
- replacement parts
- batteries
- RC transmitters & receivers

ONLINE TUTORIALS: <http://learn.parallax.com/elev-8>

PARALLAX
 www.parallax.com

Order online or call 888-512-1024 (M-F 8a-5p PDT)



FIGURE 2: The Wii™ classic controller, showing values for each button. Note that Color BASIC does not recognize the joysticks, just buttons.

by the JOY command when you press any button.

If the program doesn't seem to work when you first run it, check your cable connector (indented side up!), then reboot your system (press the reset button, or type REBOOT at the flashing cursor prompt). That should get you in business (**Figure 2**).

You should notice that the JOY command returns the value of any button pressed, or zero if no button is pressed. (See **Figure 2** for the numbers Color BASIC assigns to each button.) Also note that when two or more buttons are pressed at the same time, JOY returns the sum of their assigned values. The **a** and **b** buttons always override any other buttons, with **a** always taking priority over **b**. Also note that Color BASIC does not recognize the joysticks on the Wii controller — just the buttons — and the controller must be connected to the Amigo before the system is started.

Also, please consider using TYPE-IT.BAS as a starting point for similar programs you build, perhaps with a completely different topic.

Just LOAD "TYPE-IT.BAS," SAVE it with the name of your new program, then open the Editor and start hacking the code. Have fun, and don't forget to save your work!

2. JOY.BAS.

If you have a Wii™Classic Controller (not the nunchuck), you can use it with your Amigo to increase the number of fun things your programs can do. The secret is the Color BASIC JOY command, which works like INKEY, but for the controller buttons instead of keyboard keys.

Enter the code below, then SAVE it as "JOY.BAS" to explore this neat command:

```
10 REM *** JOY.BAS ***
20 COLOR 63,22
30 CLS
40 LOCATE 10,0: PRINT ">>>  Wii Controller
   Values <<<"
50 LOCATE 1,2: PRINT "Press Any Button, or
   SELECT+HOME to Return..."
60 REM -- Get Button Push --
70 A=JOY
80 IF A=0 THEN GOTO 70 REM <- No Button Pressed
90 REM -- Show Value --
100 LOCATE 18,18: PRINT "JOY = ";
110 PRINT A,
140 REM -- Again? --
150 IF A>12 THEN GOTO 70 REM <- SELECT+HOME?
160 CLS: END
```

Now, connect your controller to the Amigo (the indented side of the connector should face up) and RUN the program. In addition to the title and simple instructions, you should see the non-zero value returned

3. DRAW.BAS.

If you do have a Wii controller, here's a simple program snippet to make use of the JOY command. Enter this code, save it as "DRAW.BAS," then use the controller buttons to drive a little square around on the screen:

```
10 REM *** DRAW.BAS ***
20 COLOR 63,0
30 CLS
40 c=22 REM <- Color (0 to 63)
50 x=49 REM <- Graphics Column (0 to 99)
60 y=37 REM <- Graphics Row (0 to 74)
70 REM -- Get Button Push, Update Parameters --
80 IF JOY=64 AND x<99 THEN x=x+1
90 IF JOY=256 AND x>0 THEN x=x-1
100 IF JOY=128 AND y<74 THEN y=y+1
110 IF JOY=32 AND y>0 THEN y=y-1
120 IF JOY=2048 THEN c=c+1
130 REM -- Plot Point --
140 PLOT x,y,c
150 GOTO 80
```

The Color BASIC screen has 100 graphics columns, numbered 0 to 99, and 75 graphics rows, numbered 0 to 74. The PLOT command has the syntax PLOT <column>, <row>, <color>; this code simply uses button values from the controller to update where a graphics point is plotted and in what color. The result is a maze of brightly colored lines on a black screen. Like our previous typing program, at first glance this does not seem very exciting, but it's surprisingly engaging to young ones — at least until the new wears off (**Figure 3**).

If your youngster is intrigued by this little bit of code, here's a challenge from me to you; Use the main loop in this code snippet to rewrite TYPE-IT.BAS into DRAW-IT.BAS. Include controller buttons for a commands screen, changing cursor and screen colors, clearing the screen,

printing a smiley, typing a text snippet, and any other features your helper might want. I bet you'll find this easier than you think, and it should be a great joint project!

4. TIMES.BAS

Here's another "real" program to help with multiplication practice (if your student is still at that stage of his/her schooling). It's old school — not common core — but perhaps it will be helpful, and it may be something your helper can customize for a younger brother or sister. Enter this code, and save it as "TIMES.BAS:"

```
10 REM *** TIMES.BAS ***
20 COLOR 63,22
30 CLS
40 LOCATE 14,1: PRINT "~~~ Math Practice ~~~ "
50 REM -- Get Practice Range --
60 LOCATE 3,3: INPUT "First Number Low, High?
   ";c,d
70 LOCATE 3,5: INPUT "Second Number Low, High?
   ";e,f
80 LOCATE 3,8: PRINT "Press Any Key to Begin..."
85 REM -- Randomize RNG --
90 u=INKEY
100 v=RND (10)
110 IF u=0 THEN GOTO 90
115 REM -- Main Loop --
120 r=0 REM <- Correct Answers Counter
130 FOR n=1 TO 10
140 COLOR 63,22
150 CLS
160 LOCATE 14,1: PRINT "~~~ Math Practice ~~~"
170 a=RND (d-c+1)+c
180 b=RND (f-e+1)+e
190 LOCATE 20,16: PRINT a;" X ";b;" = ";
200 INPUT k
210 LOCATE 20,18
220 IF k=a*b THEN GOTO 270
225 REM -- Incorrect Response --
230 PRINT a;" X ";b;" = ";a*b
240 COLOR 20,22
250 LOCATE 20,16: PRINT a;" X ";b;" = ";k
260 GOTO 280
265 REM -- Correct Response --
270 PRINT "Correct!": r=r+1
280 PAUSE 2000
290 NEXT n
295 REM -- Again? --
300 COLOR 63,22
310 LOCATE 13,16: PRINT "You got ";r;" of 10
   correct."
320 LOCATE 20,18: PRINT "Again? (Y/N)"
330 u=INKEY
340 IF u="Y" OR u="y" THEN GOTO 120
350 IF u<>"N" AND u<>"n" THEN GOTO 330
355 REM -- New Ranges? --
360 LOCATE 18,18: PRINT "New Ranges? (Y/N)"
370 u=INKEY
380 IF u="Y" OR u="y" THEN GOTO 10
390 IF u<>"N" AND u<>"n" THEN GOTO 370
400 CLS : PRINT "Bye! Hope you had fun!"
410 END
```

When you run the code, you'll be prompted to enter the low/high ranges for the multiplier and multiplicand, and then press a key to start the drill. You'll then be presented with 10 random problems, one after another, with immediate correct/incorrect feedback after each one.

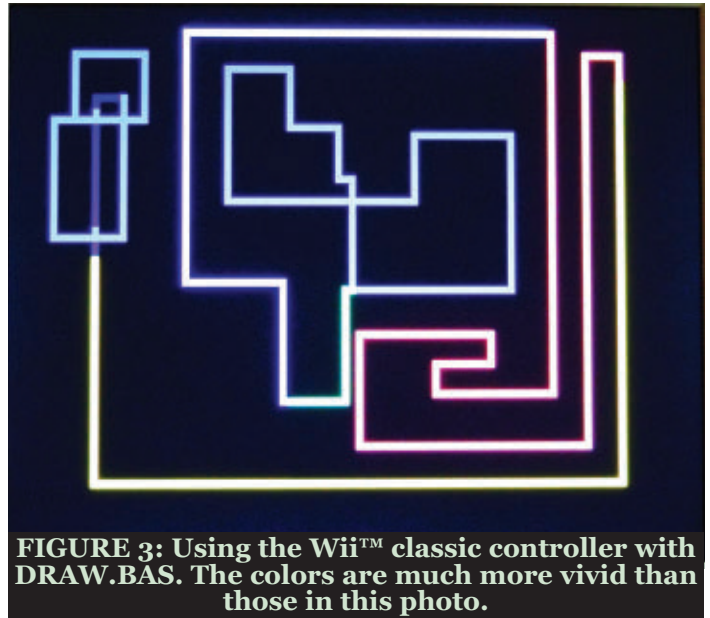


FIGURE 3: Using the Wii™ classic controller with DRAW.BAS. The colors are much more vivid than those in this photo.

You are then given the option to practice again or change the ranges.

By now in your Color BASIC journey, most of this code should be intuitive to you, but a couple of things deserve mention. Lines 85 - 110 serve to randomize the Color BASIC random number generator, which (like most software RNGs) isn't truly "random." Without this code, the random number stream after each program start-up will be the same, and problems presented for like ranges will appear in the same order each time. By using the "random" time it takes for the user to press a key, the RNG is set differently each time the program starts, with different problems for the same practice ranges.

Lines 170 and 180 pick the random multiplier and multiplicand for each problem, based on the ranges previously set. Note that the ranges can include zero on the lower end, and you can enter the same number for low and high if you want to practice a specific problem set, like "times sixes."

It will be fairly simple to modify this program for addition instead of multiplication, and subtraction and division are possible with a little thought. If you get really ambitious, you could modify the framework of TYPE-IT.BAS to include all four arithmetic operations. At any rate, I hope your student enjoys this program!

5. OPERATOR.BAS.

One of the things that can be a bit confusing when learning Color BASIC is how the various arithmetic, comparison, and logical operators work. Sooner or later, you'll bump into the limits of 32-bit integer arithmetic, or need a specific conditional test for a critical IF statement in your project. Here's an instructive little program that

lets you input two integers, and then shows you the result of each Color BASIC operation on them. Enter this code, and save it as "OPERATOR.BAS:"

```
10 REM ***** OPERATOR.BAS *****
20 COLOR 63,22
30 CLS
40 LOCATE 10,1: PRINT ">>> Color BASIC Operators
   <<< "
50 PRINT " ": INPUT " > Enter Integers A,B:
   ";A,B
```


```
60 PRINT " "
70 PRINT " A = ";A;" B = ";B
80 PRINT " ": PRINT " -- Arithmetic Operators
   -- "
90 PRINT " A+B = ";A+B
100 PRINT " A-B = ";A-B
110 PRINT " A*B = ";A*B
120 PRINT " A/B = ";A/B
130 PRINT " A//B = ";A//B
140 PRINT " ": PRINT " -- Comparison Operators
   -- "
150 PRINT " A=B = ";A=B
```

```
160 PRINT " A<B = ";A<B
170 PRINT " A>B = ";A>B
180 PRINT " A<=B = ";A<=B
190 PRINT " A>=B = ";A>=B
200 PRINT " A<>B = ";A<>B
210 PRINT " ": PRINT " --
   Logical Operators
   -- "
220 PRINT " A AND B = ";A AND
   B
230 PRINT " A OR B = ";A OR B
240 PRINT " NOT A = ";NOT A
250 PRINT " NOT B = ";NOT B
260 PRINT " ": PRINT " >>>
   Again (Y/N)?"
270 I=INKEY
280 IF I="Y" OR I="y" THEN
   GOTO 30
290 IF I="N" OR I="n" THEN END
300 GOTO 270
```

ALL ELECTRONICS

www.allelectronics.com Order Toll Free 1-800-826-5432


170 LUMEN WORK LIGHT
Nebo 6327 LarryC. EXTREMELY BRIGHT! 170 lumens of intense light. 180° rotating magnetic clip. Water and impact-resistant. Powered by 3 AAA batteries (included). **\$8.99 each**
CAT# FL-170



18 AWG RED/BLACK ZIP CORD
Automotive grade "GPT" zip cord with red and black conductors. 80°C, flame-retardant PVC insulation. Available in rolls of 25, 50, 100, 250, 500 and 1000 feet. We will cut to order, 10 ft. minimum. **25' for \$8.00**
CAT# WRB-18
50' for \$15.00 • 100' for \$23.00




3W STEREO AMPLIFIER MODULE
High-quality 3W stereo amplifier based on the PAM8403 chip. Low current drain. Operates on 2.5-5Vdc. 24.5 x 15.0 x 3.5mm pc board. **\$4.00 each**
CAT# AMP-8403



COLOR HEAT SHRINK TUBING ASSORTMENT
154 pieces. Unbreakable clear plastic box with dividers contains 4 inch lengths of 6 different diameters:
1/16" - 56 pcs
3/32" - 35 pcs
1/8" - 28 pcs
3/16" - 18 pcs
1/4" - 9 pcs
3/8" - 8 pcs
\$11.95 each
CAT# HS-4900




12VDC 3A POWER SUPPLY, USED
Input: 100-120 Vac
Output: 12V, 3A
6' output cable w/ 2.5mm coax plug, center+. cULus. Used, guaranteed functional. **\$7.75 each**
CAT# PS-123U
10 for \$7.25 each



Reduced Price! POWER STRIP FOR 19" RACK
Chatsworth Products # 712828-409. Rack-mount 8-outlet power strip. 10' cord w/ standard NEMA 5-15 grounded plug & sockets. 12A 125V 50/60Hz. Power indicator lamp. cULus. **\$18.95 each**
CAT# PST-8409 ~~\$22.00~~



12VDC ULTRABRIGHT WHITE LED, 5MM
Water-clear 5mm package. 12 Vdc, built-in resistor. Operates at lower voltages with lower brightness. **50¢ each**
CAT# LED-12WP
10 for 45¢ each • 100 for 40¢ each



ON-OFF TOGGLE SWITCH, 15A 120 VAC
S.P.S.T., on-off switch. 15/32" threaded bushing mount. Screw terminals. **\$1.85 each**
CAT# STS-110
10 for \$1.75 each



When you run this program, you should see a prompt to enter two integers. When you do, the program lists the results of each Color BASIC arithmetic, comparison, and logical operation on those integers. The five arithmetic operations always return a 32-bit integer value.

The six comparison operations always return either a -1 (the comparison is TRUE) or 0 (the comparison is FALSE). The three logical operators of NOT, AND, and OR also return either a -1 or a 0 value.

Note that Color BASIC supports bitwise operators, but they will be discussed in a future article and are not covered here.

So, that completes five more easy projects for the Mentor's Friend! I hope your Amigo and Color BASIC are bringing big smiles to you and any young helpers, and that you are starting a list of projects to tackle with this cool little platform.

Future Mentor's Friend articles will discuss some simple (and fun!) hardware interface projects for your Amigo.

Until then, enjoy! **NV**

Introducing the ME Labs Tutorial and Trainer Board

This package includes everything you need to learn to program PIC® Micro microcontrollers: tutorial, board, software, and micro USB cable.



Special Price of \$149.99

Use Coupon Code **NV516** at checkout

Valid only until May 30, 2016 (Retail price \$189.99)

Academic and bulk pricing available

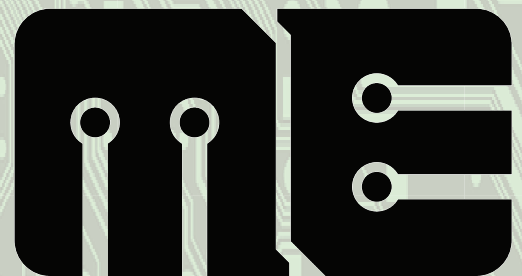
- Learn to write code and program PIC® microcontrollers with one of the most complete programming tutorials available!
- Learn the fundamental concepts of programming, including program flow, loops, coding techniques, binary manipulation, device-to-device communication, user interface design, in-circuit debugging, and more!
- If you've dabbled with other languages and want to fully learn the extensive capabilities of PBP in a wide range of applications, this is for you!
- 14 Lessons use a building-block approach with specific exercises designed to help you master each topic.

ME Labs has been a global leader in the creation of development tools for PIC® microcontrollers for 20 years. To learn more and to order go to: <http://melabs.com> or contact us at: info@melabs.com

Check out our other popular products for PIC developers including:

U2 Programmer
PBP 3.0 Compiler
Prototyping Boards
Experimenter Boards

PBP3
PICBASIC PRO™ Compiler



L • A • B • S
www.melabs.com

PICBASIC PRO is a trademark of Microchip Technology Inc. in the USA and other countries.
PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

Portable RF Mesh Alarm System Can be Monitored from Anywhere

Back in the day of the dinosaur, an alarm system was defined as a man or woman standing guard at the entrance of the home cave. At the first sign of danger, the prehistoric “alarm system” would sound off, alerting all of those in ear shot. Caveman alarms became more sophisticated when the guards communicated with each other using smoke, drums, or runners. Things haven’t changed much. Even with today’s high tech security systems, there are still men and women “standing guard” in certain environments. This month’s installment of Design Cycle will focus on deploying some high tech security devices as tools for the watchmen.

A40 Mesh Networking Alarm Controller

The A40 Mesh Networking Alarm Controller is a smart RF node that can be deployed in an alarm network of up to 40 A40 nodes (one master/39 slaves). The A40 you see in **Photo 1** is driven by a PIC16F883 microcontroller and can be powered with a nine volt C battery pack. The PIC’s job is to encode and decode RF traffic that is driven among the A40 nodes participating in the alarm mesh network. Each A40 node is addressed using hex and BCD rotary switches.

A node address consists of a Site Address (1-15) combined with a Unit ID (0-39). The site master node resides at Unit ID 0. The master A40 is no more than a normal A40 mesh networking alarm controller that generates the synchronization signals for the slave units operating within a particular site.

As you can see in **Figure 1**, every node in an A40 alarm network is electronically identical in its base construction. A typical

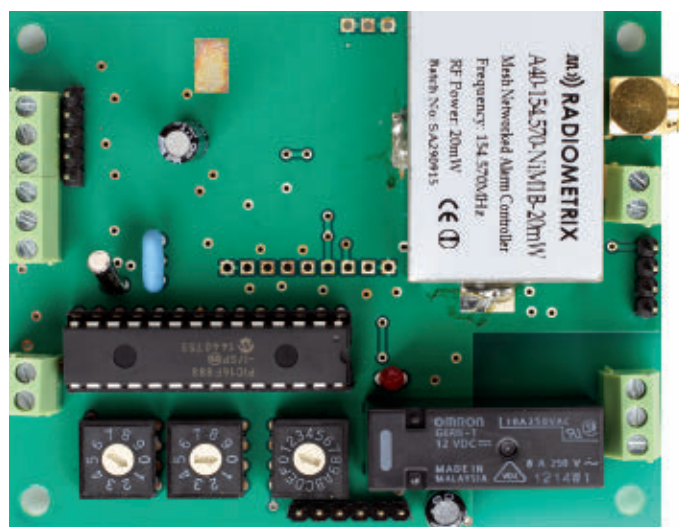
A40 node consists of an RF transceiver, a PIC, three address switches, a relay status LED (D16), a MOSFET switch, and an SPDT relay. Each A40 node also has the capability of driving an optional tri-color LED which visually indicates certain network status information. The MOSFET is intended to drive a siren device. The relay can switch loads that do not exceed 8A at 250 VAC or 5A at 30 VDC. A relay status LED provides visual indication of the relay state.

In addition to the pair of MOSFET and relay outputs, the A40 is equipped with three inputs. The A40 ALARM, TEST, and CANCEL inputs are active low logical inputs that can be implemented with simple pushbuttons or normally open contacts. A40 node status is available in serial format via the A40 alarm controller’s STAT output pin. Low battery voltage is indicated by the BATL output which emits 100 ms pulses every four seconds once the battery falls below the low voltage threshold.

Satellite nodes can be attached to normal A40 nodes by placing a jumper between the A40’s SAT pin and GND. Satellite nodes are configured with the same Unit ID of their associated normal A40 node, and are only able to transmit to the associated normal A40 node with the identical Unit ID. There is no limit to the number of satellite nodes that can be attached to a network. However, adding more than 10 will result in RF traffic congestion. This is

not a serious problem since the satellite units retransmit their activation states using a pseudorandom dithered rule. Configuring an A40 mesh networking alarm controller with a Unit ID of 40 will force the A40 node into Drone mode. Drones are listen-only nodes.

If you take an A40 alarm controller node for what it is physically, all we have is a pushbutton-controlled radio that can sense an alarm input and control a couple of output devices. If your view of the mesh networking alarm controller is logical, you see the ability



■ **PHOTO 1.** The A40 Mesh Networking Alarm Controller is a smart RF node that is designed to be used in temporary and permanent alarm solutions.

to replace the pushbuttons and monitor the alarm network status via the A40's STAT and BATL pins using an external microcontroller and companion touch panel.

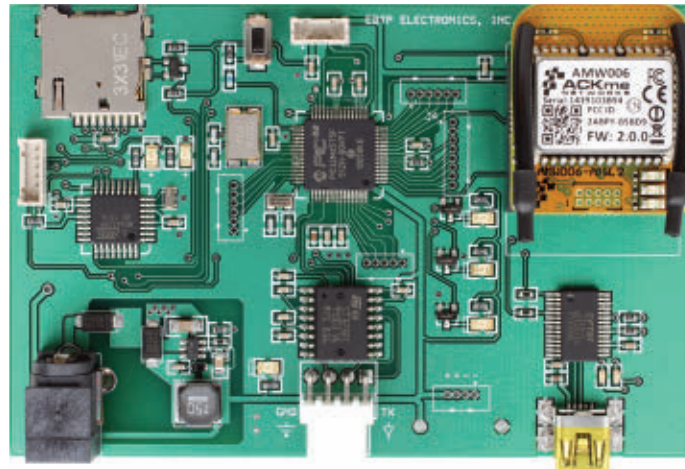
Helper Hardware

An active alarm input on any of the nodes in an A40 alarm network will trigger the alarm circuitry in all of the nodes that are active on the network. So, we will need to add circuitry and firmware that will inform us as to which node initiated the alarm.

Fortunately, each A40 node provides its status to the master node every 32 seconds. All we have to do is capture and decode the A40 response packet. Response time for an alarm is not 32 seconds. The A40 network has provisions to pass along an active alarm condition to a peer node within four seconds. The discovering A40 alarm outputs are activated immediately.

The primary communications link for any A40 node is the RF channel. Other than visual indicators, a typical A40 node will only communicate status information via its STAT pin, which is transmit only. So, despite the need for a receive-only serial port configuration, our external microcontroller hardware will be required to provide two complete serial ports. One of the external microcontroller's serial ports will be used to service the A40's STAT pin and a second serial port will communicate with the touch panel.

Since we're expending the energy to generate hardware and firmware to assist our A40 alarm network, we might as well plan ahead. Depending on the alarm solution, it may become necessary to log network activity. So, we'll include a microSD storage



■ **PHOTO 2.** As luck would have it, we have already designed, tested, and written driver code for our A40 mesh networking alarm controller hardware helper. The addition of the computer and communications power of this little board puts the A40 alarm controller into the IoT space.

subsystem in our hardware design to support that possibility. There may also be requirements to monitor and control the A40 network remotely. IoT (Internet of Things) technology can make an A40 network available

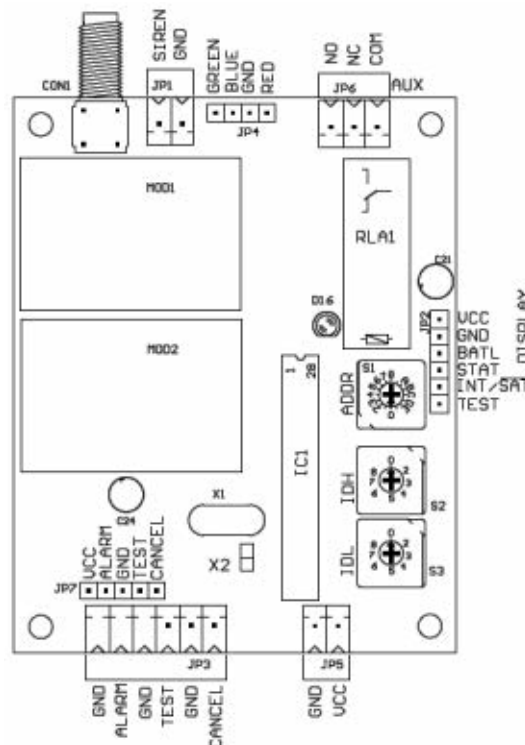
to other embedded devices, as well as smartphones and tablets. For that to become feasible, we'll need to design in an Internet-capable radio system. Since USB is everywhere these days, USB will also be represented in our helper hardware.

The multiple serial port and microSD requirements of our A40 helper hardware design scream PIC32. It just so happens that we have already designed and tested our A40 helper hardware. Break out your July 2015 copy of *Nuts & Volts*. In that edition of Design Cycle, we

designed and built a universal communications board that was based on the Microchip PIC32MX575F512H. At the time, we did not take into consideration that we were really designing and building an IoT device.

Our reconstituted IoT device is reprised in **Photo 2**. Logging and non-volatile data storage capabilities are provided by the OpenLog based microSD subsystem. The true RS-232 serial port we will use to interface the touch panel is based on an ST3232C. The XBee-compatible pair of 10-pin female headers can be used by Wi-Fi and Bluetooth radio devices. Incoming data on any of the serial interfaces can be processed and passed to any other of the helper hardware's serial interfaces.

The USB portal is capable of being logically attached to the microSD card. This logical connection allows the user to access the microSD card from a terminal session running on a laptop computer. Our helper hardware is also equipped with an RTCC (Real Time Clock Calendar), three user-programmable LEDs, and DF-13 male I/O headers. In that we covered design theory and construction in the July 2015



■ **FIGURE 1.** The A40 mesh networking alarm controller is laid out to accommodate pushbuttons as input devices. However, the A40 controller's onboard PIC allows us to replace the pushbuttons with mechatronic logic.

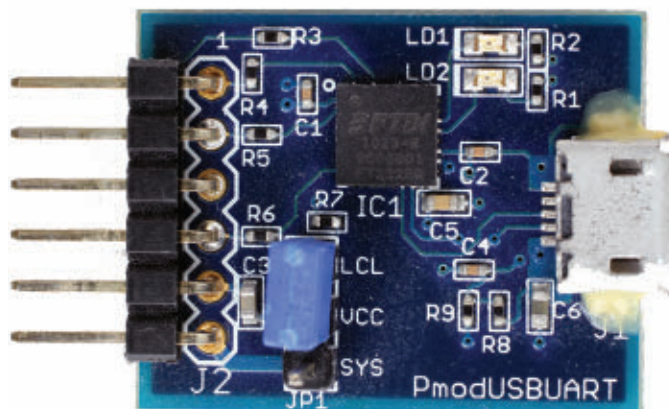
Byte	Value	Description
1	0-15	System Address
2	0-39 255	Normal Unit ID Drone
3		Network status record byte
Bit 7	1 = Low	Local low battery
Bit 6	1 = Low	Network low battery
Bit 5	0	
Bit 4	0	
Bit 3	0	
Bit 2		Cancel mode (note: cancel over-rides alarm, which over-rides test)
Bit 1		Alarm mode (if bit 2 is high, bits 1,0 have no meaning)
Bit 0		Test mode (if bit 1 is high, bit 0 has no meaning)
4-13		Burst payload bytes. Bit-level coded exactly as the radio link burst format
14		Current sync slot
15		Current sync frame (low 4 bits) and Zone (high 4 bits)
16		Heartbeat number (cycles 0-255, incrementing on each burst sent)

■ **FIGURE 2.** No rocket science here. However, what this chart doesn't show is the A40 node address information that is encoded within the burst payload bytes.

edition of Design Cycle, we won't go there in this installment. However, I will provide a complete set of ExpressPCB files for the helper hardware via the article link.

Decoding the A40 Status Packet

The A40 STAT pin emits 16 bytes of status information in a 3.3 volt serial RS-232 format. What this means is that the A40 STAT serial stream is compatible with 3.3 volt microcontroller UARTs. The A40 serial format allows us to easily tap into the STAT data stream using a Digilent PmodUSBUSART module (Photo 3) and a laptop PC running CCS's Serial Input/Output Monitor program, which comes as part of the CCS C compiler package. If you don't own a copy of the CCS C compiler, you can utilize any terminal emulator program that displays incoming data in hexadecimal format.



■ **PHOTO 3.** This is the most versatile little module on the EDTP bench.

Figure 2 lays out the definition of the bytes contained within the A40 16-byte status message. That's nice, but we need to know a bit more. I have a pair of A40 units, so let's configure them and fire them up and see what we get.

I'll set up one A40 unit as the master by setting both of its Unit ID rotary switches to zero (00). I'll set the master A40 Site Address rotary switch to 1. I'll also set the Site

Address rotary switch of the A40 slave unit to 1. We'll begin by addressing the A40 slave with a Unit ID of 01. We now have a master A40 unit operating on site 1 and an A40 slave unit operating on site 1 with a Unit ID of 01. The Pmod USB module's RXD pin is attached to the STAT pin of the master A40. A laptop running the CCS serial I/O monitor is attached to the USB end of the Pmod USB module. Once I apply power to both the A40 master and slave, we will expect to see a 16-byte response come in from the A40 slave in about 32 seconds. Here is what came in:

```
01 00 00 00 00 00 00 00 00 00 00
00 00 05 00 00 01
```

From left to right, we know the first byte is the site address. The second byte is the local Unit ID, which for the master A40 node is 00. According to the third byte, everything is lovely as there are no

■ **FIGURE 3.** Here is a compilation of all 39 address encodings. As you would imagine, the Heartbeat, Site Address, and Network Status fields will reflect the actual network configuration and state.

```
** A40 Slave Unit ID Decode
** Heartbeat values vary and are consecutive for clarity
** Site Address range 01-15
01 00 00 00 00 00 00 00 00 00 00 00 00 05 00 00 01 - Slave Unit ID = 01
01 00 00 00 00 00 00 00 00 00 00 00 00 11 00 00 02 - Slave Unit ID = 02
01 00 00 00 00 00 00 00 00 00 00 00 00 41 00 00 03 - Slave Unit ID = 03
01 00 00 00 00 00 00 00 00 00 00 00 00 01 01 00 00 04 - Slave Unit ID = 04
01 00 00 00 00 00 00 00 00 00 00 00 00 04 01 00 00 05 - Slave Unit ID = 05
01 00 00 00 00 00 00 00 00 00 00 00 00 10 01 00 00 06 - Slave Unit ID = 06
01 00 00 00 00 00 00 00 00 00 00 00 00 40 01 00 00 07 - Slave Unit ID = 07
01 00 00 00 00 00 00 00 00 00 00 00 01 00 01 00 00 08 - Slave Unit ID = 08
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 01 00 00 09 - Slave Unit ID = 09
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 01 00 00 0A - Slave Unit ID = 10
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 01 00 00 0B - Slave Unit ID = 11
01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 01 00 00 0C - Slave Unit ID = 12
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 01 00 00 0D - Slave Unit ID = 13
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 01 00 00 0E - Slave Unit ID = 14
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 01 00 00 0F - Slave Unit ID = 15
01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 01 00 00 10 - Slave Unit ID = 16
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 01 00 00 11 - Slave Unit ID = 17
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 01 00 00 12 - Slave Unit ID = 18
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 01 00 00 13 - Slave Unit ID = 19
01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 01 00 00 14 - Slave Unit ID = 20
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 01 00 00 15 - Slave Unit ID = 21
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 01 00 00 16 - Slave Unit ID = 22
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 01 00 00 17 - Slave Unit ID = 23
01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 01 00 00 18 - Slave Unit ID = 24
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 01 00 00 19 - Slave Unit ID = 25
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 01 00 00 1A - Slave Unit ID = 26
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 01 00 00 1B - Slave Unit ID = 27
01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 1C - Slave Unit ID = 28
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 01 00 00 1D - Slave Unit ID = 29
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 01 00 00 1E - Slave Unit ID = 30
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 01 00 00 1F - Slave Unit ID = 31
01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 20 - Slave Unit ID = 32
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 01 00 00 21 - Slave Unit ID = 33
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 01 00 00 22 - Slave Unit ID = 34
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 01 00 00 23 - Slave Unit ID = 35
01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 24 - Slave Unit ID = 36
01 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 01 00 00 25 - Slave Unit ID = 37
01 00 00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 01 00 00 26 - Slave Unit ID = 38
01 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 01 00 00 27 - Slave Unit ID = 39
```


alarms and no low batteries. The next byte we know about for sure is the last one, which says this is heartbeat number 0x01. What the heck is that 0x05 byte? To find out, I'll switch the A40 slave unit's Unit ID to 02 and power-on reset the slave A40. We'll wait for 32 more seconds and see what comes in. Tick ... Tick ... Tick ... Here's the next packet:

```
01 00 00 00 00 00 00 00 00 00
00 00 11 00 00 02
```

The first and last bytes are exactly what we would expect. However, our 0x05 has changed to 0x11. This must mean that the responding A40 slave's Unit ID is 02. Let's switch the slave A40 to Unit ID 03 and make an observation:

```
01 00 00 00 00 00 00 00 00 00
00 00 41 00 00 03
```

So far, we can't come to any logical conclusions about the relationship between 0x05, 0x11, and 0x41. All we can do is keep the packets coming in and noting the changes to the packet values versus the Unit ID setting of the A40 slave. So, let's examine some more packets. I'll add a Unit ID notation at the end of each packet for our reference. Ignore the final heartbeat byte of each packet as I may be starting and stopping the A40 master during this experimentation process:

```
01 00 00 00 00 00 00 00 00 00
00 01 01 00 00 14 - Slave Unit
ID = 04
01 00 00 00 00 00 00 00 00 00
00 04 01 00 00 15 - Slave Unit
ID = 05
01 00 00 00 00 00 00 00 00 00
00 10 01 00 00 16 - Slave Unit
ID = 06
01 00 00 00 00 00 00 00 00 00
00 40 01 00 00 17 - Slave Unit
ID = 07
```

Hey, looky here! We finally can see a logical pattern. Hopefully, we'll see something we can latch onto with the next four packets:

```
01 00 00 00 00 00 00 00 00 00
01 00 01 00 00 07 - Slave Unit
ID = 08
01 00 00 00 00 00 00 00 00 00
04 00 01 00 00 08 - Slave Unit
```

```
ID = 09
01 00 00 00 00 00 00 00 00 00
10 00 01 00 00 10 - Slave Unit
ID = 10
01 00 00 00 00 00 00 00 00 00
40 00 01 00 00 12 - Slave Unit
ID = 11
```

I see a definite pattern (shifting 01 04 10 40) and I'm sure you do too. With that, I've put together a complete list of A40 slave Unit ID packet values in **Figure 3**. Now, we can single out and determine the network status of any A40 slave.

Decoding the incoming A40 node status eliminates the need for an audible alarm. Using our helper hardware, we can now "electronically" sense an A40 alarm and pinpoint its location.

Dead Easy Alarm System

I have an English friend (Trevor) that uses the description "dead easy" for things that are simple and intuitive to accomplish. Well, setting up an A40 mesh networking alarm controller network is dead easy. All you need to do to get going is to configure an A40 master by setting the A40's Unit ID rotary switches to 00. Then, select a site address for the master and dial it in. Once you have configured a master A40, then you can set up your slave A40s by dialing in the appropriate site address followed by the desired A40 Unit ID. Hook up your alarm inputs/outputs on the A40 network nodes and apply power to the A40 master and slave units. BOOM! You have configured and installed an RF mesh alarm network.

We Still Have Some Work to Do

In the next installment of Design Cycle, we will trick out our A40 mesh networking alarm controller system. We'll attach a full-color touch panel to our A40 IoT alarm hardware and write all of the code to make it go.

NV

PoLabs

For more information or software download please visit 

www.poscope.com/nv

PoKeys Connect, Control



PoKeys is a successful story about versatile and extremely powerful controller.

Save your time with fast connectivity and easy configuration of advanced features.

Enjoy rare moments in life while PoKeys is taking care over your house or garden.

Take a break while PoKeys is controlling stepper motors in your CNC or 3D printer.

Let PoKeys make your imagination come true using our intuitive software and plugins for Mach3/4

Stepper motor drivers Drive

Safest and most efficient force for your motors.



PoScope Mega1+ Measure



Do you need a really powerful and portable oscilloscope that will save your time, money and make your life with measurements easier?

PoScopeMega1+ is lowest power consumption USB oscilloscope, function generator and logic analyzer all-in-one available on the market.

GREAT READS FOR DIYers!

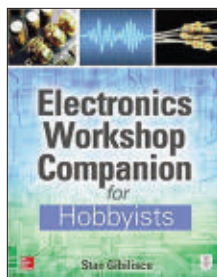
Programming the Photon: Getting Started with the Internet of Things

by Christopher Rush
Quickly learn to construct your own electronics devices and control them over the Internet with help from this DIY guide. This book features clear explanations and step-by-step examples that use inexpensive, easy-to-find components. Discover how to connect to Wi-Fi networks, attach hardware to I/O ports, write custom programs, and work from the cloud. You will learn how to troubleshoot and tweak even interface with social media sites!
\$20.00



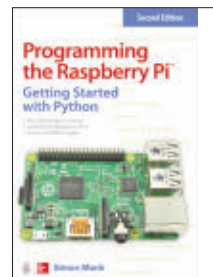
Electronics Workshop Companion for Hobbyists by Stan Gibilisco

In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets. Electronics Workshop Companion for Hobbyists provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource.
\$25.00



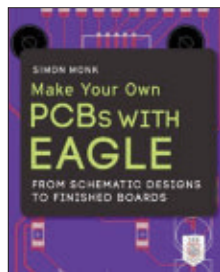
Programming the Raspberry Pi, Second Edition: Getting Started with Python

by Jack Simon
This practical book has been revised to fully cover the new Raspberry Pi 2, including upgrades to the Raspbian operating system. Discover how to configure hardware and software, write Python scripts, create user-friendly GUIs, and control external electronics. DIY projects include a hangman game, RGB LED controller, digital clock, and RasPiRobot complete with an ultrasonic rangefinder
\$15.00



Make Your Own PCBs with EAGLE by Eric Kleinert

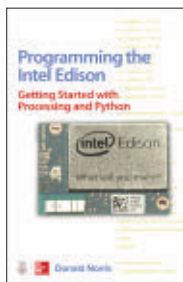
Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.
\$30.00



Programming the Intel Edison: Getting Started with Processing and Python by Donald Norris

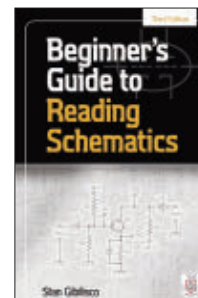
Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, WiFi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor..
\$20.00



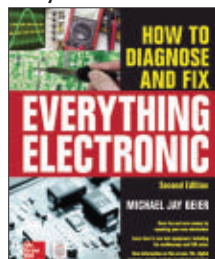
Beginner's Guide to Reading Schematics, 3E by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.
\$25.00



How to Diagnose and Fix Everything Electronic by Michael Jay Geier

**A Fully Revised
Guide to
Electronics**
Repair all kinds of electrical products, from modern digital gadgets to analog antiques, with help from this updated book. The Second Edition offers expert insights, case studies, and step-by-step instructions from a lifelong electronics guru. Discover how to assemble your workbench, use the latest test equipment, zero in on and replace dead components, and handle reassembly.
\$24.95



Programming PICs in Basic by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.
\$14.95



Programming Arduino Next Steps: Going Further with Sketches by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download.
\$20.00



Order online @ store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Beginner's Guide Educational Combo!

Price Reduced
Now Only \$219.95

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?

Joe Pardue
SmileyMicros.com

Puzzled by the Arduino?
Based on *Nuts & Volts* Smiley's Workshop, this set gives you all the pieces you need!

Educational Combo \$124.95

We have 33 amazing titles of PROJECTS FOR THE EVIL GENIUS

Need a project to help you spring into summer?
Check out the titles below

- 101 Outer Space Projects for the Evil Genius
- 101 Spy Gadgets for the Evil Genius
- 123 PIC Microcontroller Experiments for the Evil Genius
- 123 Robotics Experiments for the Evil Genius
- 126 Physics Projects for the Evil Genius
- 15 Dangerously Mad Projects for the Evil Genius
- 25 Home Automation Projects for the Evil Genius
- 30 Arduino Projects for the Evil Genius, Second Edition
- 30 BeagleBone Black Projects for the Evil Genius
- 48 Science Fair Projects for the Evil Genius
- 50 Awesome Auto Projects for the Evil Genius
- 50 Green Projects for the Evil Genius
- 50 Model Rocket Projects for the Evil Genius
- 51 High-Tech Practical Jokes for the Evil Genius
- Arduino + Android Projects for the Evil Genius
- Bike, Scooter, and Chopper Projects for the Evil Genius
- Bionics for the Evil Genius
- Electronic Circuits for the Evil Genius, Second Edition
- Electronic Gadgets for the Evil Genius 2 Ed
- Electronic Games for the Evil Genius
- Fuel Cell Projects for the Evil Genius
- Holography Projects for the Evil Genius
- Mechatronics for the Evil Genius
- Mind Performance Projects for the Evil Genius
- MORE Electronic Gadgets for the Evil Genius
- PC Mods for the Evil Genius
- PICAXE Microcontroller Projects for the Evil Genius
- Programming Video Games for the Evil Genius
- Raspberry Pi Projects for the Evil Genius
- Recycling Projects for the Evil Genius
- Solar Energy Projects for the Evil Genius
- Telephone Projects for the Evil Genius
- tinyAVR Microcontroller Projects for the Evil Genius

MORE ELECTRONIC GADGETS FOR THE EVIL GENIUS
40 NEW BUILD-IT-YOURSELF PROJECTS

Wow! Because of the great response we've had to this special, we're continuing it for one more month. All Evil Genius books on sale for \$19.95 ea for the month of May.

CD-ROM SPECIAL

Nuts & Volts 12 CD-ROMs & Hat Special!

That's 144 issues.
Complete with supporting code and media files.

Free Shipping!

Only \$249.95 or \$24.95 each.

The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise all-purpose reference featuring hundreds of tables, maps, formulas, constants & conversions. AND it still fits in your shirt pocket!

Only \$12.95

Visit store.nutsvolts.com or call (800) 783-4624

Simple. Practical. Propeller.

I am writing from the road — literally! As I type, I am a passenger in a truck heading from Knoxville to St. Louis. In my younger days, I did a lot of traveling across the country, and across the world. It was fun. I was used to it, and I knew how to travel well. I kept things simple and practical. This is a good lesson for embedded programmers of all stripes. In my job, I am often tasked with projects and schedules that are not aligned; that is to say that the project requirements far exceed the timeline requested for delivery. I had a recent incident where a client gave me barely two hours to create a reasonably sophisticated game device. We managed to pull it off, but not without keeping things simple and practical. Simplicity is elegance. Elegance is beauty, and we all love beautiful things.

I had two days in Los Angeles, CA between my current road trip and that one, and managed in that time to have a quick visit with my friend, Rick Galinson. Rick works in practical effects in the movie and TV business. He's built things like dinosaurs in the original *Jurassic Park* film, to the roller-skating ghosts in a life-sized Pac Man game for a beer commercial. By the way, those ghosts had Propellers in them running WS2812 light strips so Rick

could control their colors and the eye animations.

We stood on Rick's porch on one of the few days it's rained this year, sharing stories of mild frustration — my two hour game project in Dallas, his working with youngsters on a robotics project. Don't get me wrong, Rick loves working with kids (and adults) and is really good at it; in fact, he's an amazing teacher. What we both have noticed is that younger people seem less patient when learning new things. As adults our age do, we decided to blame the Internet. When we were kids, we couldn't look up the answer on our computer or phone. What did we do? We worked through the problem. Those of us in our age group have learned as many ways *not* to do something as to do it correctly.

Please understand: I don't claim that anything I do is particularly earth shattering. Still, when a client asks me for something, I am able to give an answer that I can stand behind — even if that answer is "I don't know." I was able to pull off the game project because of the thousands of hours of coding I've done leading up to that project, and putting those hours into a practical package that can be quickly deployed.

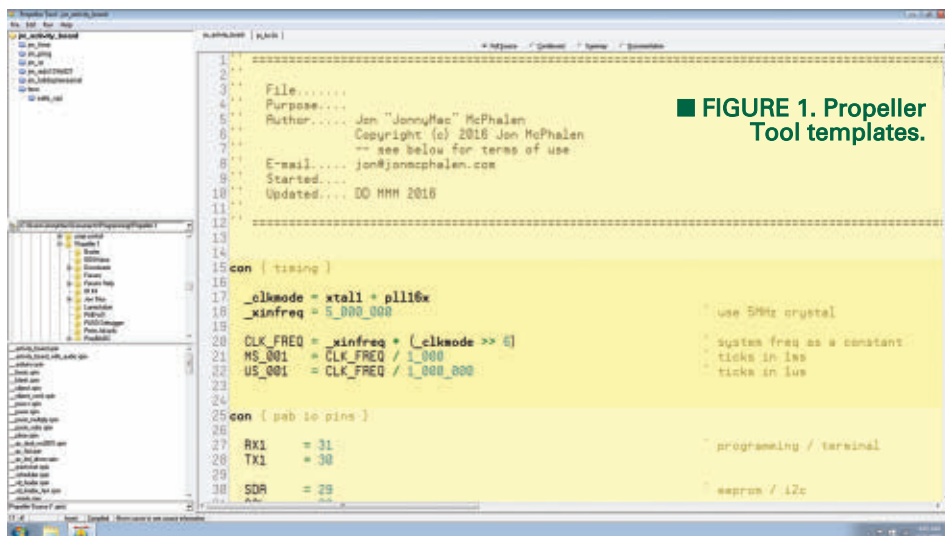
Platforms & Templates

I'm pretty lucky in that I mostly serve a specific group of customers (entertainment), and helped create a platform that I can use in a wide variety of applications: the EFX-TEK HC-8+. This board was born out of requirements that surfaced again and again — especially working with theme parks, venues like haunted houses and escape rooms, and Hollywood prop shops.

If you're focused in a particular area, you might

consider creating your own platform. Rick has his own Propeller board that satisfies the movie and TV props he works on. There is tremendous satisfaction in having one's own platform. With a wide variety of free and low cost tools, creating your own board is fairly straightforward.

Whether you create a platform or choose something that exists, the next step is to create a template for it — and then create even more. The Propeller Tool has long supported a single template file (select in the Preferences dialog), and the Propeller IDE (integrated development platform) has a facility for storing multiple templates. I'm very used to Propeller



■ FIGURE 1. Propeller Tool templates.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/May2016_SpinZone_Templates_Propellers_and_Simplicity.

Tool and like some of its editor features. To get around it only supporting one template, I preface the name of my Propeller Tool template files with two underscore characters — this forces them to be sorted to the top in the file window of the IDE (see **Figure 1**). Using this strategy, though, means I have to do an Open/Save-As sequence so that I don't overwrite my template.

The Propeller IDE has a neat new template feature that Brett Weir has been working on. Inside your **library** folder you can create a folder called **templates**. Inside this folder is where you can drop your template files, as well as an icon (PNG graphic for it).

If you're using the Prop IDE, let me suggest that you create your own library folder in your documents folder instead of using the default location. The Propeller IDE installs as a 32-bit app in 64-bit Windows systems, which means the installation folders are protected. The Prop IDE lets you select the library folder, so put this in a place where you can easily move files (not the case with protected x86 folders in 64-bit Windows systems).

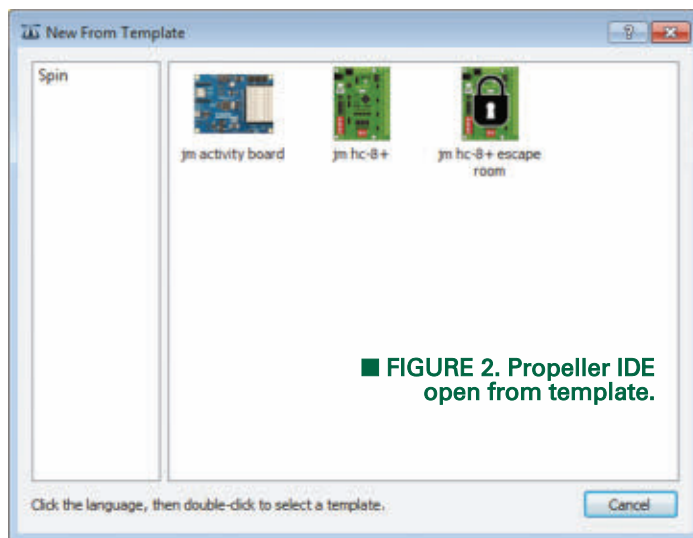
Drop your template files into the **library\templates** folder, and if you really want to fancy it up, you can also add a PNG graphic with the same name as a template. Propeller Tool will scale these for the window, so size isn't critical. I chose to go with 128x128 PNG files with an alpha channel. In **Figure 2**, you can see that I have three templates in the IDE; one for the Propeller Activity board (my go-to for experimenting), and two for the EFX-TEK HC-8+ — one of which is set up specifically for my customers in the escape room business.

I will provide you with copies of my basic templates (go to the article link), but I am begging you to write your own. Why? Because I'm just a joker for Los Angeles, and my templates may not be to your liking or be suitable for the kinds of projects that excite you. Is it work? You're darn right it is, and you should do it. Besides, it's fun!

I was having coffee one morning with John Barrowman (my partner at EFX-TEK), telling him about my conversation with Rick and my thoughts for this column. He made a really great point: *Why would anyone deny themselves the pleasure of creation?* Yes, you can find lots of code on the Internet (including mine), but you'll learn more if you create your own code. It's probably a patience thing and, interestingly, the one area of my life where I am patient. To me, coding is a lot like sculpting: You rough out the form, then — little by little — smooth out the details until it's just right. The rub is that simplicity and elegance take time, which is why I always encourage friends to spend a bit of time each day programming.

As we spend the time (when we have it), we're able to whittle away the fat of a program, or make it smarter or more efficient. There's a great quote from Blaise Pascal that speaks to this: *I have made this letter longer than usual, only because I have not had the time to make it shorter.*

Okay, enough road-warrior philosophizing. Let me show



you some of the things I've done over time to help myself with my programs.

Virtual Peripherals

As you know, the Propeller doesn't have traditional hardware peripherals like UARTs, etc.; hence, we create them virtually by launching a cog to handle those duties. The upside is that we can customize to our liking. For example, there is a four-port serial object; yes, if you need simultaneous serial streams to/from four different sources, you can have them. Remember, though, that things don't have to be this complicated to be useful. I'm not a big fan of the Arduino, but I have learned to program it so that I can help my friends. One of the things I do like about the Arduino is the **millis()** function. It returns the number of milliseconds since the last reset and is very useful for differential timing. Sure, we have the **cnt** register in the Propeller but — at 80 MHz — that rolls over in under a minute; for standardized timing, milliseconds are more useful.

Quick review: In the Propeller, we can create a synchronized loop using the **cnt** register and the **waitcnt** command. If we wanted to simulate the **millis()** function from the Arduino, we could create a little method like this:

```

pri background | t

t := cnt
repeat
    waitcnt(t += MS_001)
    ++millis
    
```

We would, of course, launch this method into its own cog, and declare **millis** in the global variable space so that any Spin cog could access it.

Neat, right? With nearly no fuss, we have a free-running counter that gives us milliseconds, and since it's a global variable, we can read it or write it from any Spin cog. Here's the rub: We're using a whole cog to increment

a variable which — by the way — consumes 336 clock cycles out of the 80,000 we have in one millisecond (assuming a 5 MHz crystal and 16x PLL setting). Okay, then, let's put that bandwidth to work. In human terms, a millisecond is no time at all — to a microcontroller, it's forever. Well, not forever, but long enough to do useful work.

Let There Be Light

The Propeller Activity board has LEDs on pins 26 and 27. These can be useful as simple status indicators. Wouldn't it be nice if we could set up a blinking state on either of the LEDs that took care of itself? You know, set and forget? Yes it would, and we have the bandwidth in the background cog to do it.

What we're going to do for those two LEDs is specify "on" and "off" times in milliseconds, then let the background code handle the switching. Here's the chunk of code that gets called from the background process loop:

```
var
    long led26on
    long led26cycles
    long led26phase

    long led27on
    long led27cycles
    long led27phase

pri update_leds
    outa[LED_26] := (led26phase < led26on)
    if (++led26phase == led26cycles)
        led26phase := 0

    outa[LED_27] := (led27phase < led27on)
    if (++led27phase == led27cycles)
        led27phase := 0
```

Each LED has variables for the on time, the total number of cycles (on and off times), and a phase variable which controls the status of the LED. Remember, the **update_leds()** method is called from our background loop so each variable is expressed in milliseconds. The first line of each section sets the LED state. If the current phase is less than the on time of the LED, the expression on the right side of the line will evaluate as true (-1, all bits set) which will cause a "1" to be written to the LED output bit. The next section increments the phase and rolls it over to zero when the cycle is complete. There is a wrapper method for setting the LEDs from the main line of our program:

```
pub blink_led(led, onms, offms)

if (led == LED_26)
    led26on := onms
    led26cycles := onms + offms
    led26phase := 0

elseif (led == LED_27)
    led27on := onms
    led27cycles := onms + offms
    led27phase := 0
```

This is simple (the way I like things); if we need an indication that causes an LED to blip for 100 ms every second, we can set it up like this:

```
blink_led(LED_26, 100, 900)
```

Boom. Bob is my uncle.

You might be wondering if we can have full on and full off with this setup. Yes, of course — just like this:

```
blink_led(LED_26, 1, 0)
blink_led(LED_27, 0, 0)
```

For on/off control, set the "off" time value to 0, and use the "on" time as a binary switch: 1 for on; 0 for off.

Finally, if we set the period short enough, the LED can appear dimmed versus blinking — this is due to the persistence-of-vision of our eyes blending the cycles together. Many suggest that 50 Hz is the minimum frequency to use so that the LED doesn't appear to flicker. In our setup, this means 20 cycles, so we don't have a lot of resolution. For simple applications (and that is our theme this month), it's enough. Here's a wrapper method for dimming:

```
pub dim_led(led, duty)

if (led == LED_26)
    led26on := duty * 20 / 100
    led26cycles := 20
    led26phase := 0

elseif (led == LED_27)
    led27on := duty * 20 / 100
    led27cycles := 20
    led27phase := 0
```

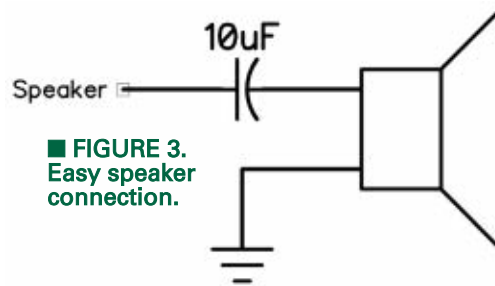
Again, not a lot of resolution, but enough to differentiate course levels; for example: off, 25%, 50%, 70%, 100%. Our eyes don't always see LEDs in a linear way, so we may at times need to create a lookup table for brightness levels.

Sounding Off

I work around entertainment and have helped create a couple of Propeller based audio boards (the EFX-TEK AP-16+ and the EFX-TEK AP-8+). That said, there are many projects where full audio is not required; simple beeps and boops work just fine. What we can do is use one of the Propeller's counters to create an oscillator and connect it to a speaker.

A note of caution: NEVER connect a speaker or any other inductive device directly to the Propeller; the surge current and inductive kickback can damage the Propeller pin. What we must do is put a buffer between the Propeller and speaker. The bare minimum is a capacitor; I use 10 μ F. It's not very loud, so not appropriate for all applications. Where it can be useful is simple audio feedback where the user is close to the device. A Propeller counter can be set into NCO mode to generate a wide variety of frequencies. That math for the counter setup is a bit hairy. If you're

interested, you can find details in the Parallax App Note (#001) on counters, and a lot of detail in the Propeller Education Kit documentation written by Andy Lindsay (the PDF is provided with Propeller Tool; if you're using the Propeller IDE, it can be downloaded from Parallax). For simple tone output, we'll specify the pin to use, the frequency, and a duration to play:



■ FIGURE 3.
Easy speaker
connection.

so let's use it. The Activity board includes a four-channel ADC (analog-to-digital converter), and my template includes a reference to an object for it. The trouble is that a clean read from the ADC requires 32 clocks and its full SPI (data going out and coming in). Doing all that in

Spin takes more time than we have. We can solve this for applications that do not require high speed reads from the ADC by breaking the read process into very small pieces; each time through the process, we'll do a small piece until the read is complete:

var

```
long  tonepin
long  tonefreq
long  tonems
```

pri play_tone

```
if (tonems > 0)
  if (ctra == 0)
    tonefreq := CLK_FREQ / tonefreq
    frqa := ($8000_0000 / tonefreq) << 1
    ctra := (%00100 << 26) | tonepin
    dira[tonepin] := 1
  else
    -tonems
  else
    ctra := 0
    outa[tonepin] := 0
```

var

```
long  analog[4]
```

```
long  astate
long  ach
long  actrl
long  awork
```

pri adc_process

```
case astate
  0 :
    outa[ADC_CS] := 0
    actrl := (ach << 27) | (ach << 11)
```

This is a one-shot player, so the process checks to see if the tone duration is greater than zero. When that is the case, it looks to see if **ctra** has been set up for the tone. If **ctra** is zero, it gets set up in NCO mode on our speaker pin, with **frqa** configured to provide the desired tone. Finally, the pin is set to output mode so we can drive the pin.

Each time through the loop, we will decrement the tone duration until it finally reaches zero; at this point, the counter is cleared which stops the tone.

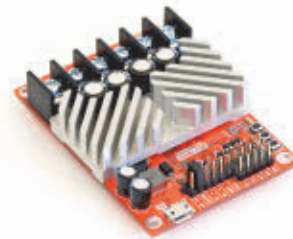
Simple. Simple. Simple. And yet, very useful. I've been doing a lot of programming for escape room puzzles and props, and many of them use buttons and switches that are nicer with additional audio feedback; a simple beep will suffice. That said, we don't always want to halt a program to play an extended beep. By using the background process, we can start the tone and then get back to work.

Of course, this idea can be extended. For one of my customers, I increased the tone resolution to 0.1 Hz and created constant definitions for musical notes. The background process uses a pointer to a table of notes and time values, and will play through them until it gets a note value less than zero; zero is used to create a rest. With this setup, we can play simple monotonic musical ditties in the background. Again, not fancy or sophisticated, but very useful in the little game prop we made for a laser tag business.

What's Your Level?

There's still some bandwidth in our background loop,

Motor Control



- 15 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial

- 45 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



- 160 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



www.ionmc.com


```

1..32 :
  outa[ADC_CLK] := 0
  outa[ADC_DI] := (actrl <= 1)
  awork := (awork << 1) | ina[ADC_DO]
  outa[ADC_CLK] := 1
33 :
  outa[ADC_CS] := 1
  analog[ach] := awork & $0FFF
  ach := ++ach & %11

if (++astate == 34)
  astate := 0

```

The variable *astate* controls the ADC read process. Each time through the ADC method, a **case** structure will allow us to run a small section of code — but not enough to break our 1 ms background process timing. At state 0, we activate the ADC by pulling the clock line low and setting the control bits for the channel. If you look at the datasheet, you'll see that you need to send 16 clocks to exchange the data. The problem is that the current 16 clocks send out the value from the last conversion, so we really want to send 32 clocks to get the current input to the channel. During states 1..32, we raise the clock line, output a bit from *actrl*, read a bit into *awork*, and then lower the clock line.

At state 33, we raise the CS pin to deactivate the ADC, clean up the result, and move it to the *analog[]* array; we then advance the channel control variable. Each time through the process, we advance the state counter and roll it over to zero. As you can see, using this state machine process to read the ADC takes time; 34 milliseconds per channel, 136 milliseconds for all. That works out to a read frequency of about 7.3 Hz. Not very fast, but this is meant to be used in those apps where we periodically read an analog input. In this case, there's no delay in accessing the ADC channel value which was read no more than 136 milliseconds ago.

Style Matters

Some may wonder why I am declaring private methods in my object — private methods are used to keep parent objects from accessing protected methods in a child object. It's just a style thing; if you find a private method in one of my top level objects, it is a warning that the method is running in another Spin cog and should not be called from the main line of the program.

I would encourage you to divide code — when you can — into small easily digested chunks. I see a lot of newcomers writing very long sections of code and getting lost in the weeds. As I am redeveloping some existing products, I find myself using a state-driven main loop that calls other methods. This allows me to focus on the structure of the program in the main loop, and flesh out the details in the methods that get called. On that thought, let's have a look at the final background control method:

```
pri background | t
```

```

io.low(LED_26)
io.low(LED_27)
io.high(ADC_CS)
io.low(ADC_CLK)
io.low(ADC_DI)
io.input(ADC_DO)

t := cnt
repeat
  waitcnt(t += MS_001)
  ++millis
  update_leds
  play_tone
  adc_process

```

This takes care of setting up the I/O pins used in the background. We have to do this from that cog so the output drivers are configured. The loop is very small, simply handling the timing and calling the methods desired in our background processes. You'll also note that I group variables for each process with the related code.

As you develop templates, consider what you have done and what you may do that will serve your customers. I have a couple of client-specific templates that are set up for me to pick and choose the elements that will be called from the background control loop.

You may be wondering how we know we've exceeded the 1 ms timing of the background loop. It's easy: After starting the background loop, we use the *millis* timer to do something. I usually just send its value to a terminal. If *millis* freezes, we've gone too far. Don't panic! Just evaluate the last set of code changes and take action. That was the case with the Activity board ADC. When I first created that code, I attempted to do the full channel read in one go. It didn't work, so I divided it into small pieces.

Funny, as I was switching back and forth between my word processor and the Propeller Tool, I found an opportunity to speed up the ADC read. I'll save that for another day. (Hint: There may be enough time to do more than one bit read in the loop timing; that is to say that it may take fewer than 32 ms to read the 32 bits.)

Keep It Simple

Regular readers know that I frequent the Propeller forums. I do that to help others and to help myself. Sometimes I'll encounter a question or solution that is very illuminating. A recent thread from a newcomer made me smile — and shake my head a bit. This person was just learning the Propeller and building a drone guidance controller at the same time. Wow! That's fine, but I would suggest that even for those of us with experience, we are best served working to the final goal slowly, in discrete easily managed chunks of code.

Keep it simple! Having eight concurrent cores doesn't mean our programs have to be complicated. In fact, we've seen this month that we can use that horsepower to simplify things. Until next time, keep spinning and winning with the Propeller! **NV**

CLASSIFIEDS

LIGHTING



LED lighting for displays, props, scenery, signs, models and more.
www.j2ledlighting.com



Automatic Stair Lighting
www.ReactiveLighting.com

Subscribe today!

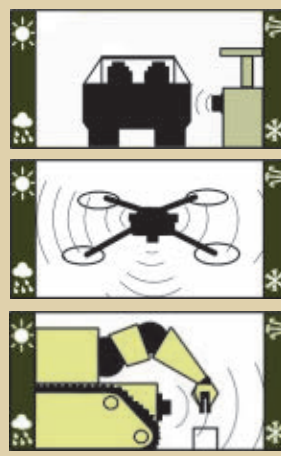
Like what you see? Go to www.nutsvolts.com

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
 Digital Equipment Corp. and compatibles.
 Buy - Sell - Trade
 CALL KEYWAYS 937-847-2300
 or email buyer@keyways.com

ROBOTICS

Need sensors?



www.maxbotix.com

EDUCATION

ELECTRONICS, MECHANICS, MATH & COMPUTING:
 Home Study Software,
 Apps & eBooks **\$2.99**
www.eptsoft.com/HomeStudy

Wanna read even more about electronics?
 Check out our book selection at
store.nutsvolts.com

LOOK FOR SEARCH FOR FIND

Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO AND TV

National RF19
 Palstar29

BUYING ELECTRONIC SURPLUS

All Electronics Corp.50

CIRCUIT BOARDS

AP Circuits34
 ExpressPCB65
 Front Panel Express LLC38
 Gooligum Electronics33
 Saelig Co., Inc.4

COMPILERS

M.E. Labs51

COMPONENTS

All Electronics Corp.50
 Electronix Express13
 NTE Parts Direct39
 Saelig Co., Inc.4
 ServoCity66

DESIGN/ENGINEERING/REPAIR SERVICES

Cleveland Institute of Electronics..34
 ExpressPCB65
 Front Panel Express LLC38
 Lemos International3
 National RF19

DEVELOPMENT PLATFORMS/TOOLS

AnarenBack Cover
 M.E. Labs51

EDUCATION

Cleveland Institute of Electronics..34

Command Productions23
 Gooligum Electronics33
 M.E. Labs51
 Poscope55

EMBEDDED SYSTEMS

Saelig Co., Inc.4
 Technologic Systems9

ENCLOSURES

Front Panel Express LLC38
 Probotix44

EVENTS

Maker Faire35

LCDs/DISPLAYS

Earth LCD17
 Newhaven Display67
 Saelig Co., Inc.4

MICROCONTROLLERS / I/O BOARDS

Gooligum Electronics33
 Ion Motion Control61
 M.E. Labs51
 Newhaven Display67
 Technologic Systems9

MISC./SURPLUS

All Electronics Corp.50
 NTE Parts Direct39

MOTORS / MOTOR CONTROL

Firgelli35
 Hitec5
 Ion Motion Control61
 Robot Power19
 ServoCity66
 The Robot MarketPlace17

PROGRAMMERS

M.E. Labs51

RF TRANSMITTERS/RECEIVERS

Lemos International3
 National RF19

ROBOTICS

Cleveland Institute of Electronics..34
 Electronix Express13
 Firgelli35
 Hitec5
 Ion Motion Control61
 Parallax47
 Robot Power19
 ServoCity66
 The Robot MarketPlace17

TEST EQUIPMENT

Electronix Express13
 NTE Parts Direct39
 Pico Technology7
 Poscope55
 Saelig Co., Inc.4
 TEquipment2

TOOLS

PanaVise38
 Poscope55
 Probotix44
 ServoCity66
 TEquipment2

WIRE, CABLE AND CONNECTORS

All Electronics Corp.50

WIRELESS PRODUCTS

AnarenBack Cover
 Lemos International3
 Technologic Systems9

All Electronics Corp.	50
Anaren	Back Cover
AP Circuits	34
Cleveland Institute of Electronics	34
Command Productions	23
Earth LCD	17
Electronix Express	13
ExpressPCB	65
Firgelli	35
Front Panel Express LLC	38
Gooligum Electronics	33
Hitec	5
Ion Motion Control	61
Lemos International	3
Maker Faire	35
M.E. Labs	51
National RF	19
Newhaven Display	67
NTE Parts Direct	39
Palstar	29
PanaVise	38
Parallax	47
Pico Technology	7
Poscope	55
Probotix	44
Robot Power	19
Saelig Co., Inc.	4
ServoCity	66
Technologic Systems	9
TEquipment	2
The Robot MarketPlace	17

>>> QUESTIONS

Low Voltage Disconnect

I am trying to build a Low Voltage Disconnect (LVD) for my 1975 Lincoln Town Car Extended Limousine. The problem is that when I do not use it daily for a longer period, the battery is discharging through some electronics. I have constructed an LVD based on two CMOS 555s connected to a cutoff relay, which is constant on until cutoff. The relay itself draws too much when in the on state. Is there a way to have a relay that draws next to nothing? It should be at least 10-16 amp contacts.

#5161

Claes Kamborn
RORVIK, Sweden

Data Transfer Dilemma

I am a mechanical engineering consultant by profession. I design the mechanical and hardware/software for my client's ideas and projects. I then 3D print the mechanical components, fabricate the PCBs, put everything together, and provide my clients with working production-intent prototypes. I love my job!

One of my elderly friends (early 80s) came to me with an antiquated product called "The Electronic Rolodex 128k." He asked me how he could get the data onto his computer because the LCD on this device is starting to fail and he is no longer able to read the names and phone numbers. I dabble enough in this area where I said that I am very capable of figuring this out. So, I took on the challenge.

The only data I/O on the device

is an IR transmitter on the left side and an IR receiver on the right. There are NO serial ports or any other connectors that would enable me to connect to a computer. The purpose of the left and right transmitter and receiver is so that if you have two identical devices, you can transmit the data from one device (on the right) to the other device (on the left).

I borrowed an IR to USB adapter from a friend and opened up a Tera Term window. I also used my signal analyzer to determine the appropriate baud rate (4800). I can see the data reception and it seems that I am receiving complete garbage characters. I have played with my signal analyzer and have inverted the signal, swapped LSB and MSB, and many other tweaks.

When that didn't work, I opened up the device and soldered leads directly to the TX and GND pins at the IR transmitter. I then connected the leads to my signal analyzer and also my oscilloscope. I verified the baud rate and also noticed that the voltage levels were TTL, not RS-232. I borrowed a TTL to USB adapter, but this did not work either.

I came to terms with the fact that the data is compressed and/or encrypted. So, I searched far and wide for some software for this device and I found it ... well at least a similar device. I installed and ran it in a variety of configurations and still cannot get receive data that makes any sense. I have done enough testing and playing around with this product where I can almost write a book, but I still have no success in transferring the data to another similar device

or to a computer. I have searched for this device on Google, eBay, Amazon, etc., and they are extinct. I've purchased a slightly newer generation of this product, but I do not believe that they are compatible. I have called the manufacturer, but they no longer support the device. So, I am at a total dead end. I have read your magazine for many years and thought that this would be the perfect venue for a *Nuts & Volts* case study/project for DIYers and the like. I would greatly appreciate any advice that you might have in doing what I originally expected to be a relatively simple problem to solve.

#5162

Neal Rosenblum
Hollywood, FL

>>> ANSWERS

[#4161 - April 2016]

Data Logger Project Guidance

I want to build two projects:

- 1.** A wireless data logger to monitor temp/humidity at various points in my home to a Windows environment.
- 2.** A wireless data logger to monitor decibels of sound. I live next to a busy street, and want to experiment with various sound suppression options and monitor success rates.

I would appreciate guidance on platforms, hardware, etc. As always, any insights and feedback are appreciated.

For the wireless networks such as yours, I recommend Digi International XBee 802.15.4 modules. They're easy to use and operate across 100 feet indoors. The XBee-PRO modules

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and

NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

have a longer range — approximately 300 feet — but they use more power. You can get longer ranges outdoors. Both operate at 2.4 GHz. The XBee modules let you transfer serial information right away. Or, you can configure them with free XCTU software from Digi to transmit API commands to remote devices and have the remote devices respond with digital and analog readings. (No code writing required.)

Internal 10-bit ADCs simplify measurement of voltages from a temperature or humidity sensor such as those sold by SparkFun or Adafruit. The XBee modules have sleep modes that save battery power. They can wake up at a given interval, transmit data, and go back to sleep. The XBee modules operate from two 1.5 volt D cells, so remote power is easy to provide.

Sound measurements require some sort of logarithmic response for decibel values. I recommend you use an LM3915 dot-bar display IC to convert sound levels to dB ranges. Most circuits use this inexpensive IC to drive 10 LEDs in a visual sound meter but you could use the eight most-significant outputs (without LEDs) to drive the eight digital inputs on an XBee module. You'll need a pull-up resistor on each of the eight LM3915 outputs. The LM3915 datasheet provides good application information and example circuits: www.ti.com/lit/ds/symlink/lm3915.pdf.

At the receiving end, you use another XBee module and connect it to your PC through a USB adapter such as the SparkFun "XBee Explorer USB" or the Parallax "XBee USB Adapter Board." A serial terminal emulator on a PC lets you see the received information.

You could write a VisualBasic program to parse the raw data from the XBee into degrees, percent

relative humidity, and decibel ranges, and then save data in a file for later analysis.

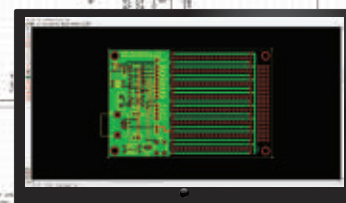
For more details, circuits, and software, I recommend, two books, "The Hands-On XBee Lab Manual," and "Wireless Sensor Networks." The

first provides information that directly relates to your needs. The latter concentrates on use of the ZigBee protocol for networks of sensors, but you must use XBee ZigBee modules.

Jon Titus
Herriman, UT

FREE

PCB Layout Software &
PCB Schematic Software



expresspcb.com



3 PCBs

\$41 + shipping

2-layer
MiniBoard

(Shipped in 1 business day)



3 PCBs

\$61 + shipping

2-layer
MiniBoardPro

(Shipped in 2 business days)



3 PCBs

\$81 + shipping

4-layer
MiniBoardPro

(Shipped in 3 business days)

- 1** **DOWNLOAD** our free CAD software
- 2** **DESIGN** your 2 or 4-layer PCB
- 3** **CHECK** your design with xCHECK DRC
- 4** **SEND** us your design with just a click
- 5** **RECEIVE** top quality boards in just days

Now Offering
**GROUND
SHIPPING**

— Continental US Customers —

LIMITED TIME PROMOTION | www.expresspcb.com

WE HAVE THE PARTS FOR YOUR IDEAS

MODIFY

USE THESE HANDY TOOLS TO CUSTOMIZE SERVOS FOR SPECIFIC APPLICATIONS



Servo Stretcher
\$19.99



HFP-25 Programmer
\$69.99



NEW DPC-11 Programmer
\$19.99



Servo Mixer \$9.95



Servo Reverser
\$39.99



Servo Power Boards
\$9.99 - \$27.99

Servo Protractor
\$17.99



ACCESSORIZE

UNIQUE AND EXCLUSIVE SERVO ACCESSORIES



Parallel Gripper Kit
\$14.99



Standard Gripper Kit A
\$9.99



Servo Gears
\$3.08 - \$14.99

Servo Sprockets

\$3.85 - \$12.99



Servo Hubs
\$3.99



Servo Arms

IMPROVE

MORE POWER
MORE STRENGTH
MORE RESOLUTION



ServoBlocks™
\$24.99 - \$28.99
PATENTED

INCREASE THE LOAD-BEARING CAPABILITIES OF YOUR SERVOS

Boosted Extensions
\$13.99 - \$15.99



Servo Gearboxes
starting at \$49.99



[Top Mount]

[Tube Mount]

[Bottom Mount]

[Channel Mount]

PRECISION CONTROL + HELPFUL FEATURES AT AN AFFORDABLE PRICE

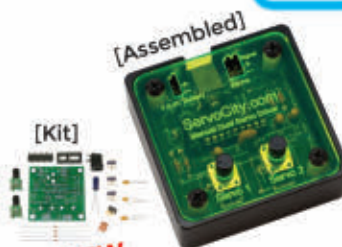
CONTROL



2 Servo Joystick
\$99.99



4 Servo Recorder
\$109.99



NEW Dual Servo Driver
\$29.99 or \$49.99



NEW Dual Servo Controller
\$59.99 or \$79.99

LET US DO THE WORK FOR YOU - WE MODIFY SERVOS!

sales@servocity.com | 620.221.0123 | www.servocity.com

Copyright 1999 - 2016 Robotzone, LLC - All rights reserved. ServoCity® and Actobotics® are registered trademarks of Robotzone, LLC

COLOR OLED Arduino Shields



OLED development made simple.



- Designed for Arduino UNO
- Breadboard friendly
- 3.3V - 5.0V ready
- Built-in logic level shifting
- Three sizes: 1.27", 1.5" and 1.69"
- Open source hardware and software
- Micro SD slot for expandable memory
- Easy to use with clearly labeled pins

NEWHAVEN DISPLAY INTERNATIONAL

For more information visit: nhdisplay.com/ArduinoShields
Phone: 847.844.8795 Email: NHsales@NewhavenDisplay.com



RoHS
Compliant

"Fetch newspaper, Rex!"



A giant, coffee-carrying IoT dinobot at the command of your index finger? Why not. In fact, wherever your imagination leads you, Anaren empowers you with the industry's easiest-to-use wireless connectivity platform. Scan the code to learn more and register for a chance to win a free Atmosphere MSDK.* Think. Build. Connect.



Anaren®
What'll we think of next?®

* Must be 18 to participate. No purchase necessary. North America only. Void where prohibited. Other restrictions apply.

www.anaren.com/airforwiced/register-win